

AD-A248 202



**DESIGN SPECIFICATIONS FOR THE ADVANCED
INSTRUCTIONAL DESIGN ADVISOR (AIDA)
(VOLUME 2 OF 2)**

Albert E. Hickey

**Mel Associates, Incorporated
1050 Waltham Street
Lexington, MA 02173**

**DTIC
ELECTE
APR 03 1992
S D D**

**J. Michael Spector
Daniel J. Muralda**

**HUMAN RESOURCES DIRECTORATE
TECHNICAL TRAINING RESEARCH DIVISION
Brooks Air Force Base, TX 78235-5000**

January 1992

Final Technical Report for Period August 1989 - July 1991

Approved for public release; distribution is unlimited.

92-08451



92 4 02 111

**AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235-5000**

ARMSTRONG

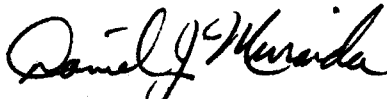
LABORATORY

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Office of Public Affairs has reviewed this paper, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This paper has been reviewed and is approved for publication.



DANIEL J. MURAIDA
Project Scientist



HENDRICK W. RUCK, Technical Director
Technical Training Research Division



RODGER D. BALLENTINE, Colonel, USAF
Chief, Technical Training Research Division

REPORT DOCUMENTATION PAGEForm Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE January 1992	3. REPORT TYPE AND DATES COVERED Final - August 1989 - July 1991	
4. TITLE AND SUBTITLE Design Specifications for the Advanced Instructional Design Advisor (AIDA) (Volume 2 of 2)			5. FUNDING NUMBERS C - F33615-88-C-0003 PE - 62205F PR - 1121 TA - 10 WU - 43	
6. AUTHOR(S) Albert E. Hickey J. Michael Spector Daniel J. Muraida				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Mei Associates, Incorporated 1050 Waltham Street Lexington, MA 02173			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES) Armstrong Laboratory Human Resources Directorate Technical Training Research Division Brooks Air Force Base, TX 78235-5000			10. SPONSORING/MONITORING AGENCY REPORT NUMBER AL-TR-1991-0085-Vol-2	
11. SUPPLEMENTARY NOTES Armstrong Laboratory Technical Monitor: Daniel J. Muraida, (512) 536-2981				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This is the final report for the second phase effort on the Advanced Instructional Design Advisor (AIDA) project. An experimental system called XAIDA is described. The proposed XAIDA is intended to assist subject-matter experts in the design and development of computer-based instructional materials. The functional requirements for an automated and intelligent advisor that would be appropriate for Air Force technical development are presented, along with a system specification.				
14. SUBJECT TERMS automated instruction computer-based instruction courseware design			15. NUMBER OF PAGES 136	
instructional development interactive technology			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

APPENDIX A

LIST OF DOCUMENTS PRODUCED IN TASK 0013

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



APPENDIX A

LIST OF DOCUMENTS PRODUCED IN TASK 0013

<u>Description</u>	<u>Date</u>	<u>Pages</u>
<u>Administration</u>		
<u>AF Task Description/Specification, Task Order 0013</u> <u>Title: Design Specifications for an Advanced</u> <u>Instructional Design Advisor (AIDA) for CBT</u>		5
<u>Task Order Proposal, Mei Associates, Inc.</u>	29 Jan 90	20
<u>Agenda</u>		
Agenda: Kick-off Meeting	26 Apr 90	1
Agenda: First Tech. Interchange Mtg.	1-2 Aug 90	2
Agenda: First Working Group Mtg.	4-6 Sep 90	2
Agenda: Second Tech. Interchange Mtg.	7-8 Nov 90	3
Agenda: Final Working Group Mtg.	14-16 Jan 91	3
<u>Task Assignments, Cycle 1</u>		
Halff, H.M.	27 Jun 90	2
Polson, M.C.	27 Jun 90	2
Reigeluth, C.M.	27 Jun 90	2
Tennyson, R.D.	27 Jun 90	2
<u>Task Assignments, Cycle 2</u>		
Halff, H.M.	7 Oct 90	1
Reigeluth, C.M.	27 Sep 90	1
Merrill, M.D.	3 Oct 90	1
Polson, M.C.	3 Oct 90	1
<u>AIDA Phase II Technical Reports</u>		
Halff, H.M. Automating Maintenance Trg.	Jul 90	28
Halff, H.M. Teaching Procedures	Nov 90	27
Halff, H.M. Authoring Facilities for Maint. Trg.	Dec 90	7
Merrill, M.D. ITX for Maint. Trg.	Sep 90	20
Merrill, M.D. & Jones, M.K. RAPIDS/TRX Walk-through	Oct 90	54
Reigeluth, C.M. Epitomizing the Experimental AIDA: What Should the XAIDA include?	30 Jul 90	5

APPENDIX A (CONT'D)

Polson, M.C. & Polson, P.G.	AIDA: Procedural Training	Jul 90	21
Tennyson, R.D.	Prototype AIDA: A Modified ISD Model	Jul 90	32
Polson, M.C.	Task Analysis in AIDA	Nov 90	12
Towne, D.M.	RAPIDS Walk-through	Nov 90	44

Technical Notes

Hickey, A.E.	Transcript of the TIM-1 90-day plan	Aug 90	17
Hickey, A.E.	Proceedings of the Working Group Meeting at USU 4-6 Sep 90	18 Sep	8
Hickey, A.E.	Proceedings of the Working Group Meeting at San Antonio, TX	14-16 Jan 91	3

AIDA Phase II Progress Reviews - J.M. Spector

No. 1	9 May 90	38
No. 2	29 Aug 90	17
No. 3	18 Dec 90	16
No. 4	21 Mar 91	15

APPENDIX B
AIDA STORYBOARD PROTOTYPE

AIDA	Help	Analysis	TAHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

**Stepping Through
the
AIDA**
Advanced Instructional Design Advisor
Storyboard Prototype

 8 November 1990

 Prepared for:
 AIR FORCE HUMAN RESOURCES LABORATORY/IDC
 Brooks AFB, TX 78235

MEI ASSOCIATES INC. #000. AIDA Paper-Based Walk-Through

AIDA	Help	Analysis	TAHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA
Advanced Instructional Design Advisor

 Please enter:
 Last Name:
 First Name:
 Middle Initial:
 Social Security Number: ____-____-____

MEI ASSOCIATES INC. #001. Initial sign-on screen

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
Version	<div style="text-align: center;"> <h1>AIDA</h1> <h2>Advanced Instructional Design Advisor</h2> <p>Please enter:</p> <p>Last Name: _____</p> <p>First Name: _____</p> <p>Middle Initial: _____</p> <p>Social Security Number: ____-____-____</p> </div>					
Control Panel						
Clock						
Calculator						
Network						
Telecom						
Quit						

MEI ASSOCIATES INC. #002 Initial sign-on screen with AIDA menu pulled down

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
	Guided Tour	<div style="text-align: center;"> <h1>AIDA</h1> <h2>Advanced Instructional Design Advisor</h2> <p>Please enter:</p> <p>Last Name: _____</p> <p>First Name: _____</p> <p>Middle Initial: _____</p> <p>Social Security Number: ____-____-____</p> </div>				
	Advice					
	JITT					
	Why?					
	On-Line Docs					
	Index					
	Return					

MEI ASSOCIATES INC. #003 Initial sign-on screen with Help menu pulled down

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
		Student				
		Environment				
		Task				

AIDA
Advanced Instructional Design Advisor

Please enter:
 Last Name:
 First Name:
 Middle Initial:
 Social Security Number: ____-____-____

MEI ASSOCIATES INC. # 004. Initial sign-on screen with Analysis menu pulled down.

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
		TRHS Library				
		Authoring				
		Configuring				
		Detailing				
		Creating				

AIDA
Advanced Instructional Design Advisor

Please enter:
 Last Name:
 First Name:
 Middle Initial:
 Social Security Number ____-____-____

MEI ASSOCIATES INC. # 005. Initial sign-on screen with TRXS Mgr menu pulled down

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
				Course		
				Enterprise TRH		

AIDA
Advanced Instructional Design Advisor

Please enter:
 Last Name:
 First Name:
 Middle Initial:
 Social Security Number: ____-____-____

MEI ASSOCIATES INC. #006 Initial sign-on screen with Design menu pulled down

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
					Course	
					TRH Enterprise	
					Simulation	
					TRH Shell	
					SME/ID	
					Student	
					Class	

AIDA
Advanced Instructional Design Advisor

Please enter:
 Last Name:
 First Name:
 Middle Initial:
 Social Security Number: ____-____-____

MEI ASSOCIATES INC. #007 Initial sign-on screen with Evaluate menu pulled down

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
						Text
						Point
						Draw
						Scan
						Audio
						CD-ROM

AIDA

Advanced Instructional Design Advisor

Please enter:
 Last Name:
 First Name:
 Middle Initial:
 Social Security Number: ____-____-____

MEI ASSOCIATES INC. #008 Initial sign-on screen with Tools menu pulled down

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

Advanced Instructional Design Advisor

Please enter:
 Last Name: Expert
 First Name: Steven
 Middle Initial: M
 Social Security Number: 999-99-9999

MEI ASSOCIATES INC. #009 SME logs in his full name and social security number

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

Advanced Instructional Design Advisor

Please click button to identify yourself.

Student

SME/ID

Visitor

Quit

MEI ASSOCIATES INC. # 010. User type identification screen. User selects "SME/ID"

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

Advanced Instructional Design Advisor

Please enter password:

MEI ASSOCIATES INC. # 011. AIDA requests password

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

Advanced Instructional Design Advisor

Valid logon:
Steven M. Expert as SME/ID
on 8 November 1990
at 0830 hours.

MEI ASSOCIATES INC. # 012. AIDA confirms valid logon.

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

Student
Environment
Task

AIDA

Advanced Instructional Design Advisor

Valid logon:
Steven M. Expert as SME/ID
on 8 November 1990
at 0830 hours.

MEI ASSOCIATES INC. # 013. SME pulls down Analysis menu and selects "Student"

AIDA	Help	Analysis	TRIS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

Student Parameter Entry

The typical student's motivation in this course is:

MEI ASSOCIATES INC. #014. SME is asked to select parameter for student motivation

AIDA	Help	Analysis	TRIS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

Guided Tour
Advice
JITT
Why?
On-Line Docs
Index
Return

AIDA

Student Parameter Entry

The typical student's motivation in this course is:

MEI ASSOCIATES INC. #015. SME pulls down Help menu and selects "Advice"

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

Student Parameter Entry

Motivation Mini-Expert

Why will the student take this instruction?

Volunteer

The student will have chosen to take it.

Required

The student will be required to take it.

MEI ASSOCIATES INC. # 016 Context-sensitive mini-expert for "Motivation" is invoked.
SME selects "Volunteer."

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

Student Parameter Entry

Motivation Mini-Expert

Will this instruction contribute to a job promotion?

Promotion

The instruction will contribute to a job promotion.

Unrelated

The instruction will not directly contribute to a job promotion.

MEI ASSOCIATES INC. # 017 Mini-expert for "Motivation" continues to gather information
SME selects "Promotion"

AIDA	Help	Analysis	TRANS Mgr	Design	Evaluate	Tools
------	------	----------	-----------	--------	----------	-------

AIDA

Student Parameter Entry

Motivation Mini-Expert

Will this instruction contribute to a pay raise?

Pay Raise

The instruction will make the student eligible for a pay increase.

Unrelated

The instruction will not directly contribute to a pay increase.

MEI ASSOCIATES INC. # 018 Mini-expert for "Motivation" continues to gather information
SME selects "Pay Raise"

AIDA	Help	Analysis	TRANS Mgr	Design	Evaluate	Tools
------	------	----------	-----------	--------	----------	-------

AIDA

Student Parameter Entry

Motivation Mini-Expert

Will this instruction enable or require a job change?

Job Change

The instruction will enable or require a job change

Unrelated

The instruction will not directly contribute to a change in job

MEI ASSOCIATES INC. # 019 Mini-expert for "Motivation" continues to gather information
SME selects "Job Change"

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

Student Parameter Entry

Motivation Mini-Expert

Did the student request a job change which requires this instruction?

Job Change

The student has requested a job change which requires this instruction.

Unrelated

The student has not made a specific request for a job change.

No Job Change

The student has requested not to change to the job which requires this instruction.

MEI ASSOCIATES INC. # 020 Mini-expert for "Motivation" continues to gather information
SME selects "Job Change"

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

Student Parameter Entry

Motivation Mini-Expert

Motivation

is prescribed to be

High

Unrelated

The student has not made a specific request for a job change

No Job Change

The student has requested not to change to the job which requires this instruction

MEI ASSOCIATES INC. # 020 Mini-expert prescribes "Motivation" as "High" on basis of SME's responses

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

Guided Tour

Advice

JITT

Why?

On-Line Docs

Index

Return

AIDA

Student Parameter Entry

Motivation Mini-Expert

Motivation

is prescribed to be

High

Unrelated

The student has not made a specific request for a job change.

No Job Change

The student has requested not to change to the job which requires this instruction.

MEI ASSOCIATES INC. #022 SME pulls down Help menu and selects "Why?"

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

Student Parameter Entry

Motivation Mini-Expert

Why is "Motivation" prescribed to be high?

Because

- The student has requested a job change which requires the instruction,
- and
- The instruction will enable or require a job change;
- and
- The instruction will make the student eligible for a pay raise;
- and
- The instruction will contribute to a job promotion,
- and
- The student will have chosen to take it

MEI ASSOCIATES INC. #023 AIDA steps back through the chain of reasoning that led the "Motivation" mini-expert to prescribe that "Motivation" is "High"

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

Guided Tour

Advice

JITT

Why?

On-Line Docs

Index

Return

AIDA

Student Parameter Entry

Motivation Mini-Expert

Why is "motivation" prescribed to be high?

Because

- The student has requested a job change which requires the instruction;
- and
- The instruction will enable or require a job change;
- and
- The instruction will make the student eligible for a pay raise;
- and
- The instruction will contribute to a job promotion;
- and
- The student will have chosen to take it.

MEI ASSOCIATES INC. #024 SME pulls down Help menu and selects "Return."

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

Student Parameter Entry

The typical student's motivation in this course is:

High

Medium

Low

Return

MEI ASSOCIATES INC. #025 SME is again asked to select parameter for student motivation SME selects "High" and returns to top level screen

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
		Student	<div>AIDA</div> <div>Advanced Instructional Design Advisor</div>			
		Environment				
		Task				

MEI ASSOCIATES INC. # 026. SME pulls down Analysis menu and selects "Environment"

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
<div>AIDA</div> <div>Environment Parameter Entry</div> <div>Click on environment factor for parameter entry:</div> <div> <div>Abstraction Levels</div> <div>Fidelity</div> <div>Sequences</div> <div>Help Routines</div> <div>Return</div> </div>						

MEI ASSOCIATES INC. # 027 SME is asked to select environment factor for parameter entry. Instead, SME elects to "Return" to top level screen

AIDA	Help	Analysis	TRMS Mgr	Design	Evaluate	Tools
		Student	AIDA			
		Environment				
		Task				

Advanced Instructional Design Advisor

MEI ASSOCIATES INC. #028. SME pulls down Analysis menu and selects "Task."

AIDA	Help	Analysis	TRMS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

Task Analysis

Click on task analysis approach:

Discursive

Graphical

Return

MEI ASSOCIATES INC. #029. SME is asked to select task analysis approach. This time, SME selects "Discursive" approach first.

AIDA	Help	Analysis	THHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

Task Analysis

Discursive

LAPS: Logic Aids for Problem Solving

Click button to select session:

Case

Dechunk

Alternatives

Return

MEI ASSOCIATES INC. • 030. SME encounters LAPS main menu.

AIDA	Help	Analysis	THHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

Guided Tour

Advice

JITT

Why?

On-Line Docs

Index

Return

AIDA

Task Analysis

Discursive

LAPS: Logic Aids for Problem Solving

Click button to select session:

Case

Dechunk

Alternatives

Return

MEI ASSOCIATES INC. • 031. SME seeks HELP and selects Guided Tour

RTDR	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

Guided Tour to LAPS

1 Overview

LAPS uses a case-based approach to decompiling the expertise of the SME, interleaving

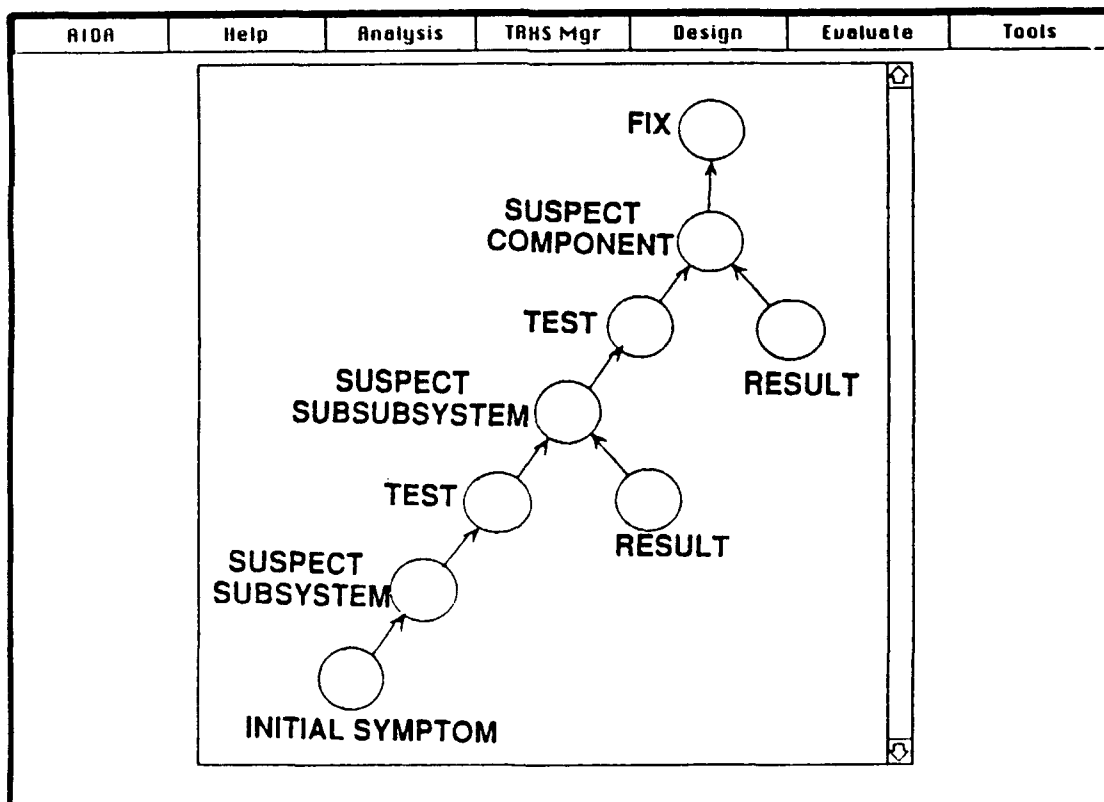
- knowledge acquisition (gathering),
- knowledge structuring (modelling),
- knowledge testing (validation and verification), and
- knowledge encoding

The current version of LAPS involves 3 sessions:

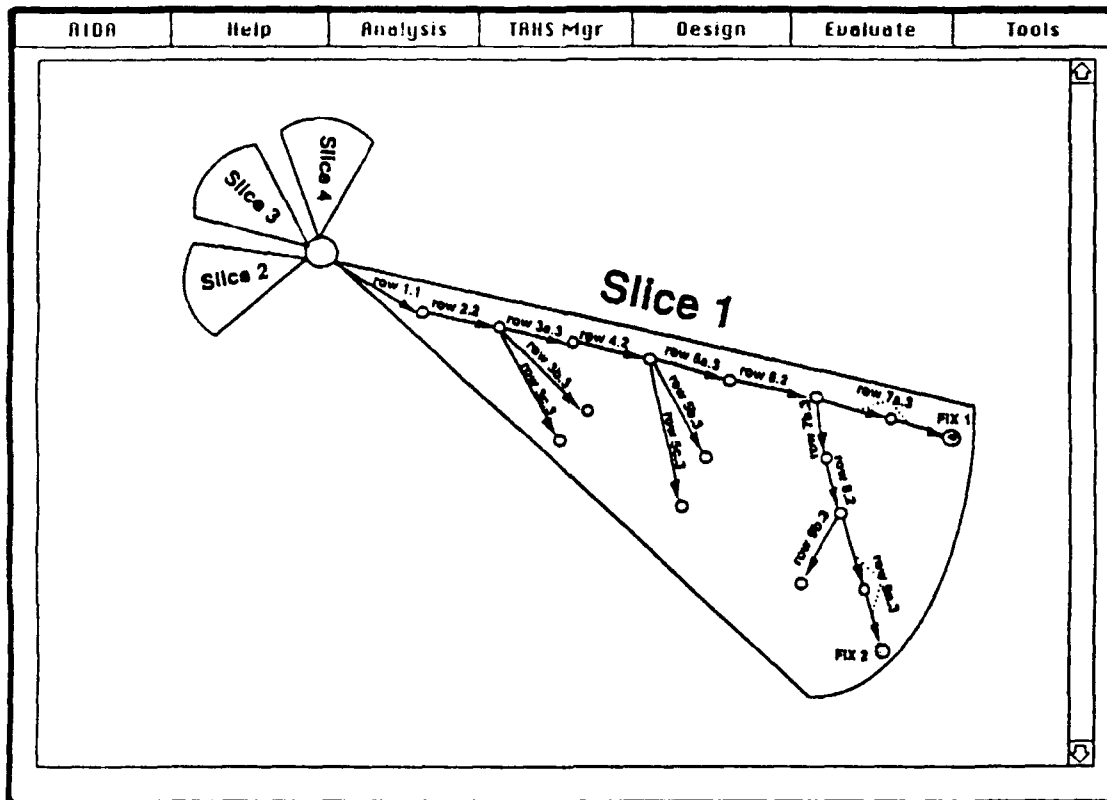
#1 The first, or case, session extracts one or more solved cases, or sample solution paths. This "sample solution path" consists of starting facts or usually followed by test results and least some reasoning steps or intermediate conclusions that the expert takes en route to his final conclusion. Associated with each statement is a list of reasons that, as we see, is in the form of numbers for statements already occurring on the list. The expert is free to run as many paths as he desires, although he usually describes only a few cases -- some are routine and others difficult.

#2 The second session "dechunks," or extracts steps that the user-expert might have omitted in the original solution path.

MEI ASSOCIATES INC. # 032. Guided Tour begins with Overview



MEI ASSOCIATES INC. # 033. LAPS Guided Tour includes graphics. Inference Tree of Attributes



MEI ASSOCIATES INC. • 034. LAPS Guided Tour. More graphics: Completion Strategy

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
	Guided Tour	<p>its modified depth-first approach to developing the necessity and sufficiency testing of the rows or rules, induction and dechunking techniques to arrive at a attributes, and its aids to logical thinking and base bookkeeping. LAPS is much more than a acquisition spreadsheet. Moreover, at every turn in tions help the user to test the rule base for y and completeness, and there are other knowledge-prodding guided-induction questions. Accordingly, the presence of these queries supports the thesis underlying laps that the following knowledge-messaging operations should be interwoven into one fabric -- the</p> <ul style="list-style-type: none"> • gathering, • structuring, • testing, <p>and</p> <ul style="list-style-type: none"> • encoding <p>of the SME's knowledge</p> <p>In conclusion, advanced instructional design advisory systems are climbing beyond their current productivity plateau, where almost all commercial knowledge engineering tools are still first first-generation. LAPS provides some of the missing components, namely, a verbal knowledge acquisition front end and automatic encoding</p>				
	Advice					
	JITT					
	Why?					
	On-Line Docs					
	Index					
	Return					

MEI ASSOCIATES INC. • 035. SME ends Guided Tour by selecting "Return" from Help Menu

AIDA	Help	Analysis	TAHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

Task Analysis

Discursive

LAPS: Logic Aids for Problem Solving

Click button to select session:

Case

Dechunk

Alternatives

Return

MEI ASSOCIATES INC. # 036. SME selects "Case" session

AIDA	Help	Analysis	TAHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

Task Analysis

LAPS: Logic Aids for Problem Solving

Click button to select specific case:

T-38A Starting System

Desert Shield Desalination Still

TBD

TBD

Return

MEI ASSOCIATES INC. # 037. SME selects specific case session for "Desert Shield Desalination Still"

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

Task Analysis

Click on task analysis approach:

Discursive

Graphical

Return

MEI ASSOCIATES INC. # 046. SME is asked to select task analysis approach. This time, SME selects "Graphical" approach.

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

Task Analysis

RAPIDS II

Click button to select specific simulation:

T-38A Starting System

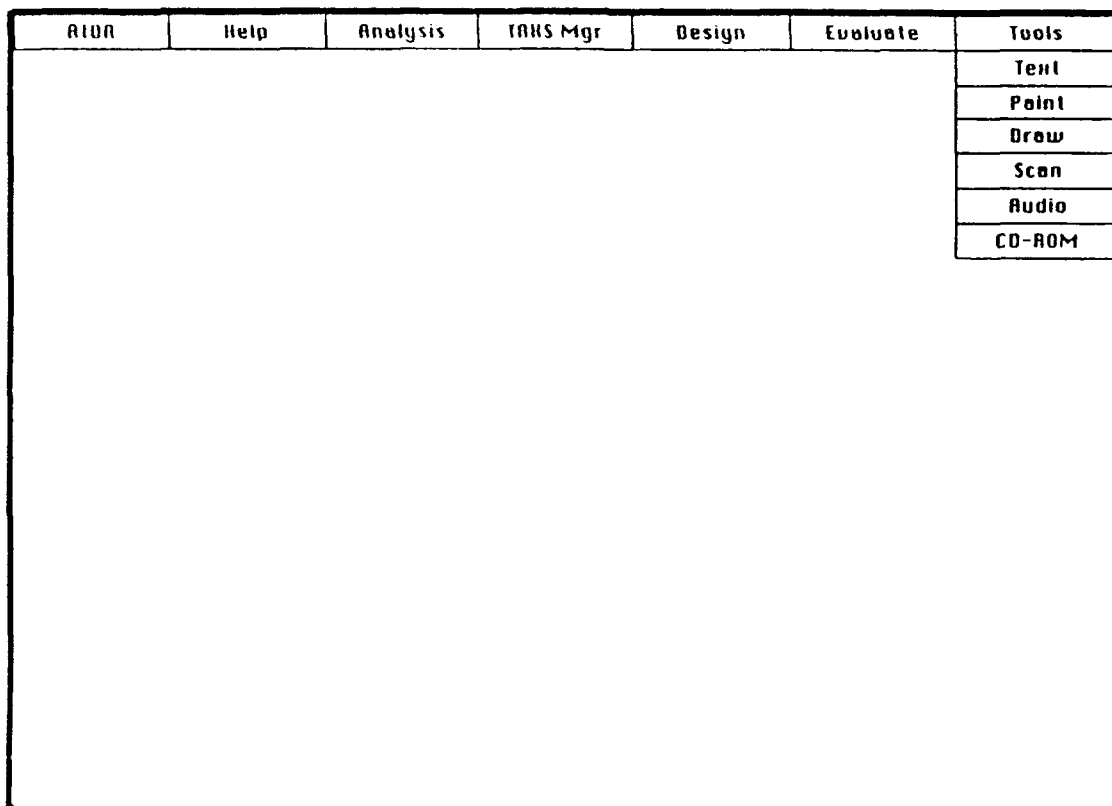
Desert Shield Desatination Still

T80

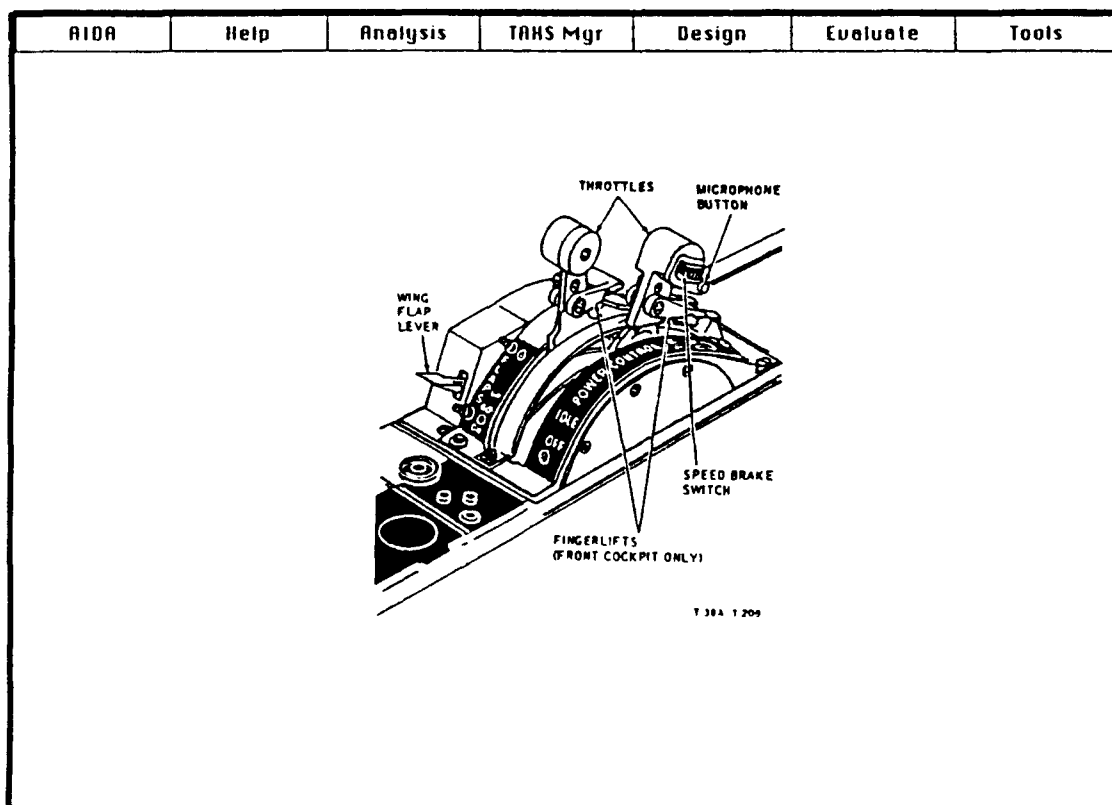
T80

Return

MEI ASSOCIATES INC. # 048. SME selects specific simulation for "T-38A Starting System"



MEI ASSOCIATES INC. #049. SME pulls down Tools menu and selects "Scan."



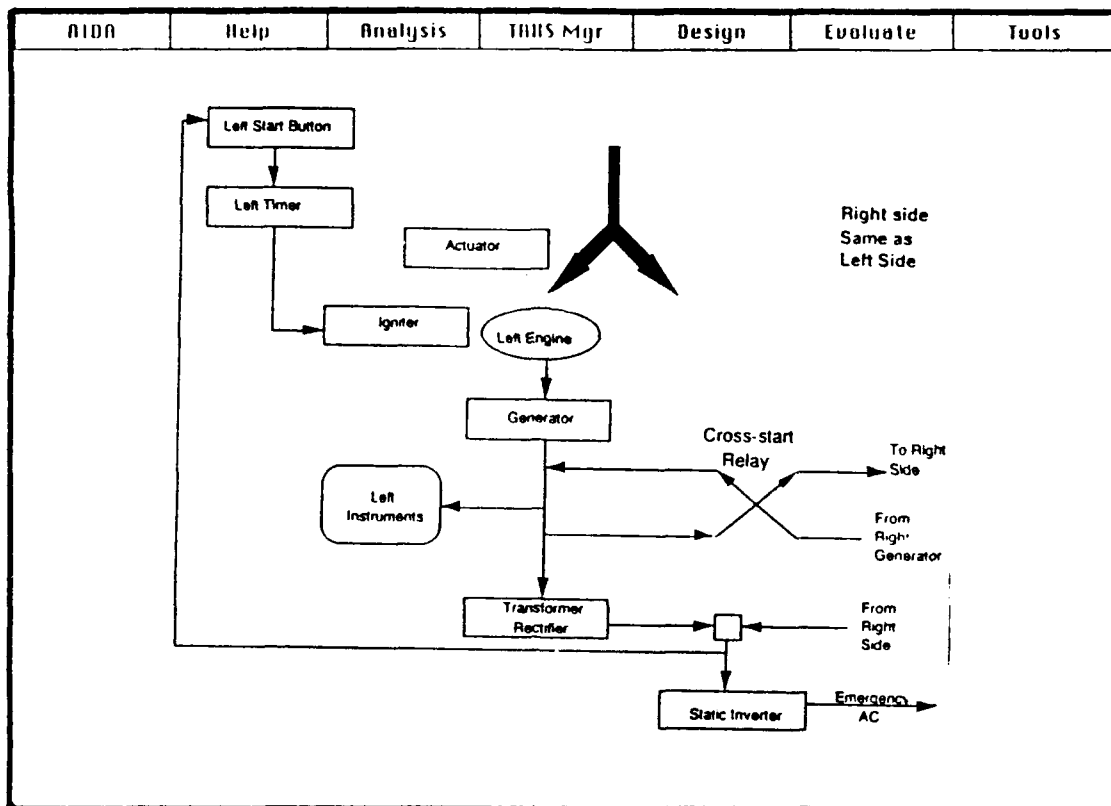
MEI ASSOCIATES INC. #050. SME scans in graphic of T-38A Throttle for physical model

RTDN	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
						Text
						Paint
						Draw
						Scan
						Audio
						CD-ROM

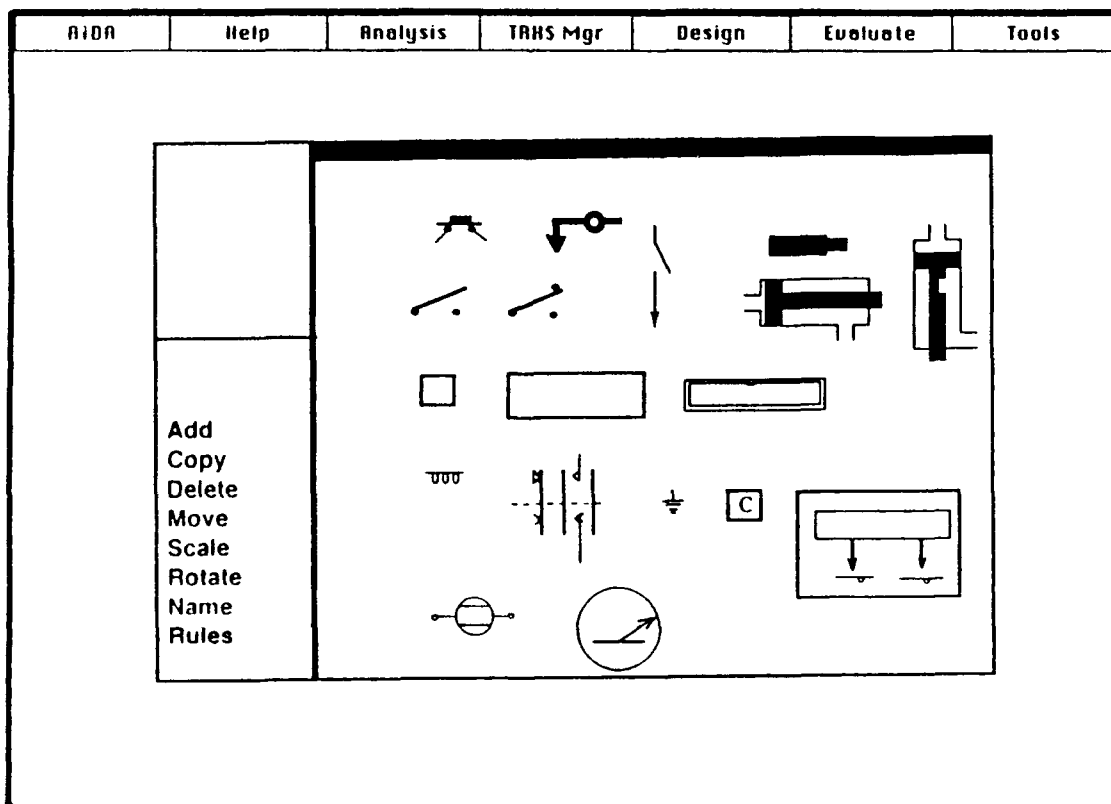
HEI ASSOCIATES INC. #051. SME pulls down Tools menu and selects "Draw."

RTDN	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
						Guided Tour
						Advice
						JITT
						Why?
						On-Line Docs
						Index
						Return

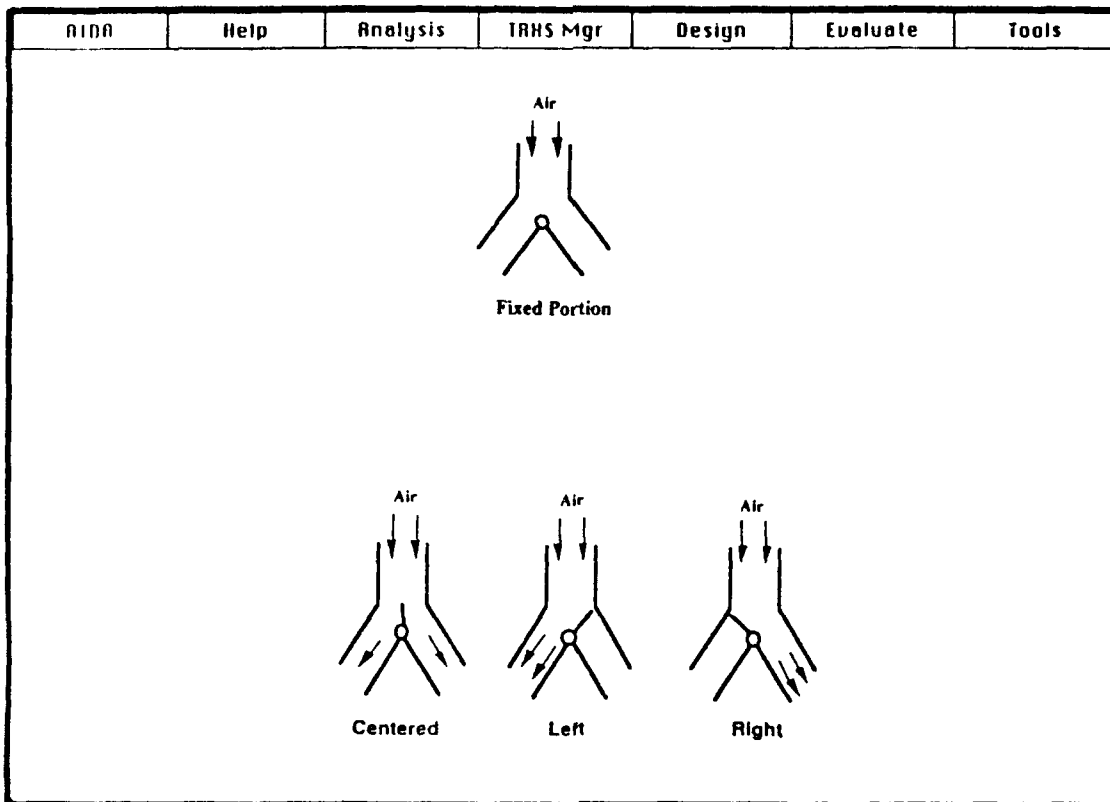
HEI ASSOCIATES INC. #052. SME pulls down Help menu and selects "JITT" for "Just-in-time Training" review on how to use mouse to draw diagrams



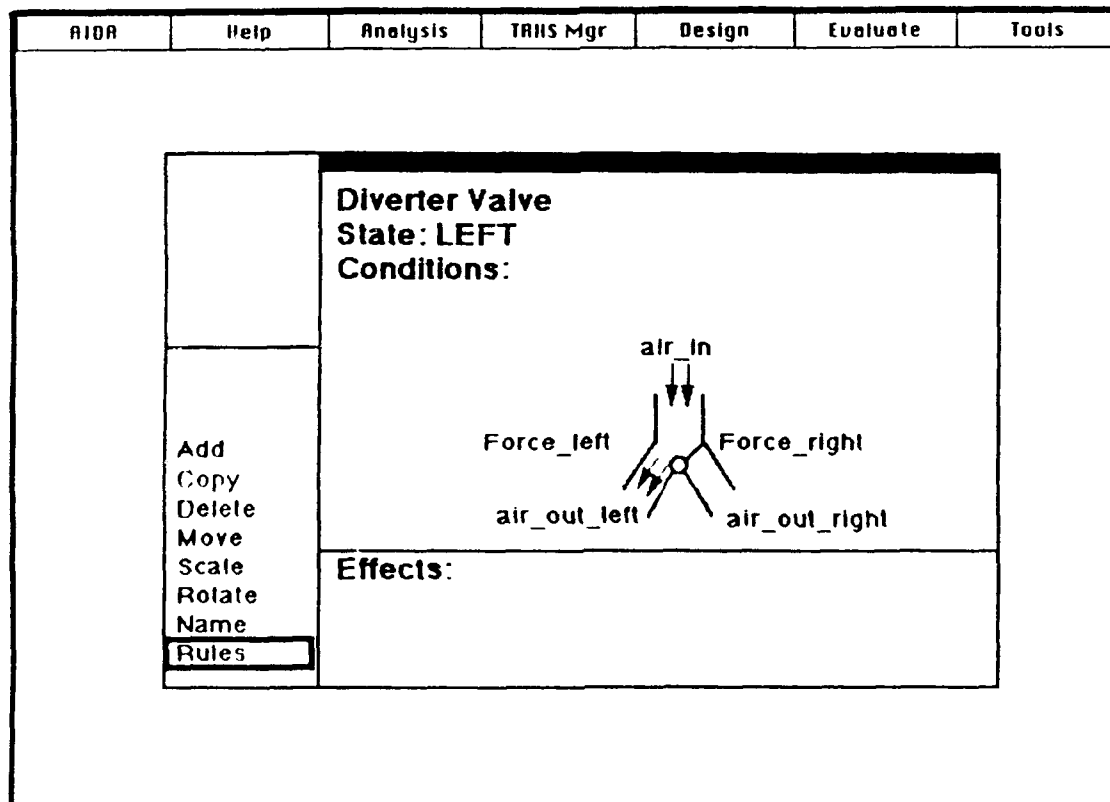
HEI ASSOCIATES INC. # 053. SME draws preliminary sketch of detailed schematic scene



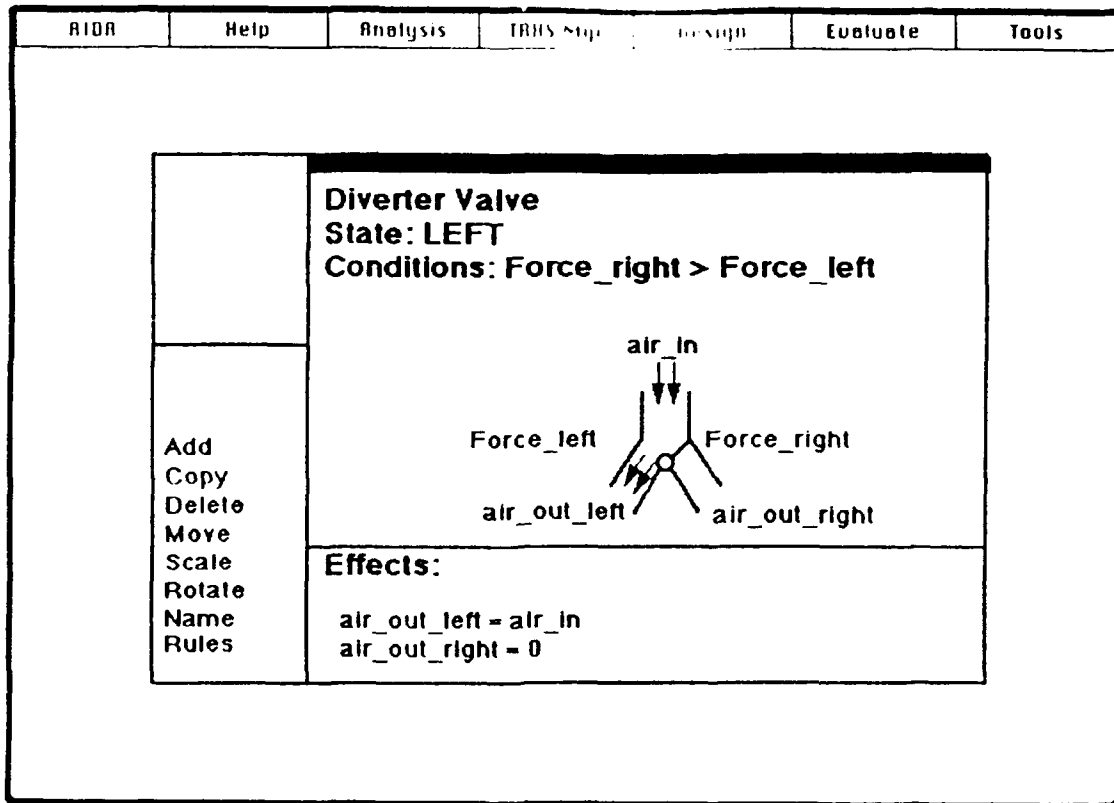
HEI ASSOCIATES INC. # 054. SME examines portion of RAPIDS II object library



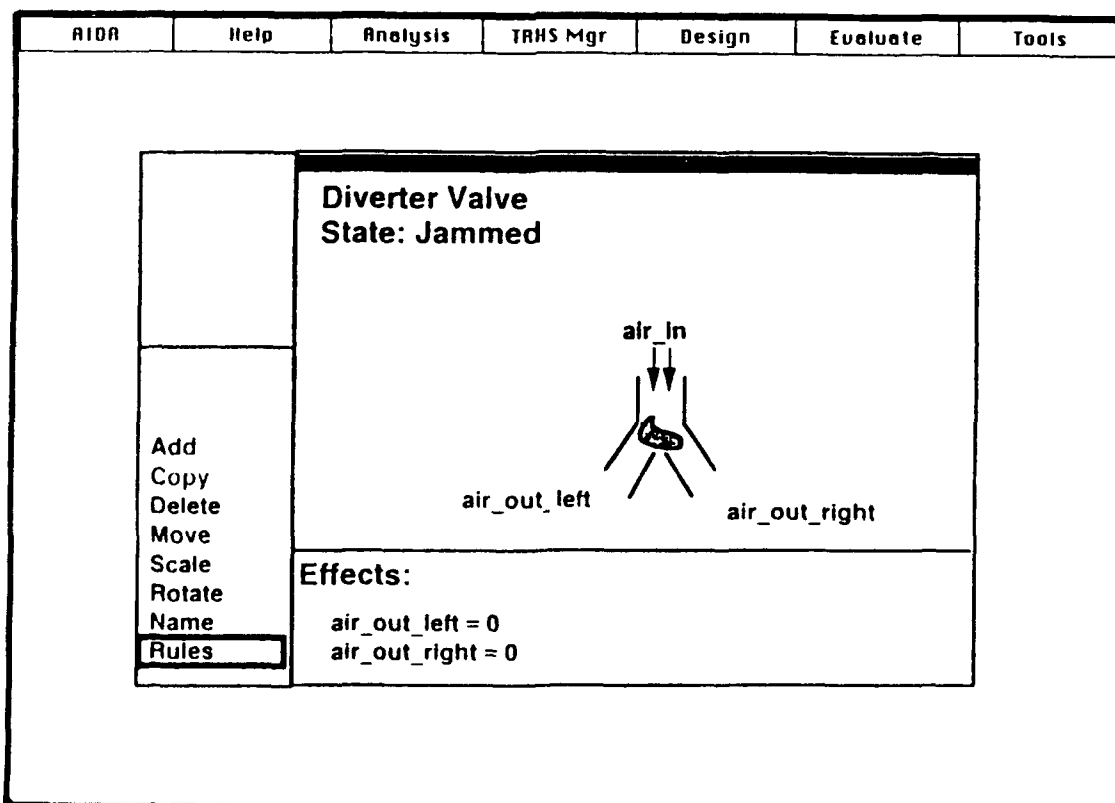
MEI ASSOCIATES INC. #055. SME draws a new functional object.



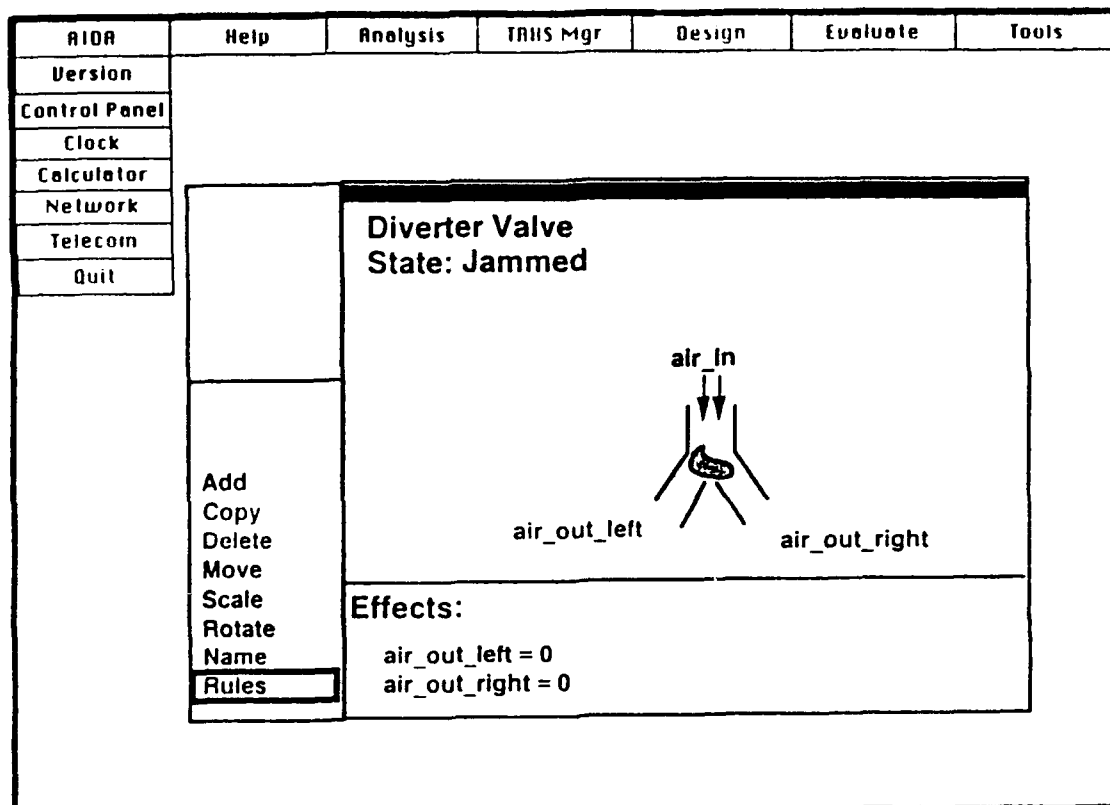
MEI ASSOCIATES INC. #056. SME starts to define rules for the Diverter Valve



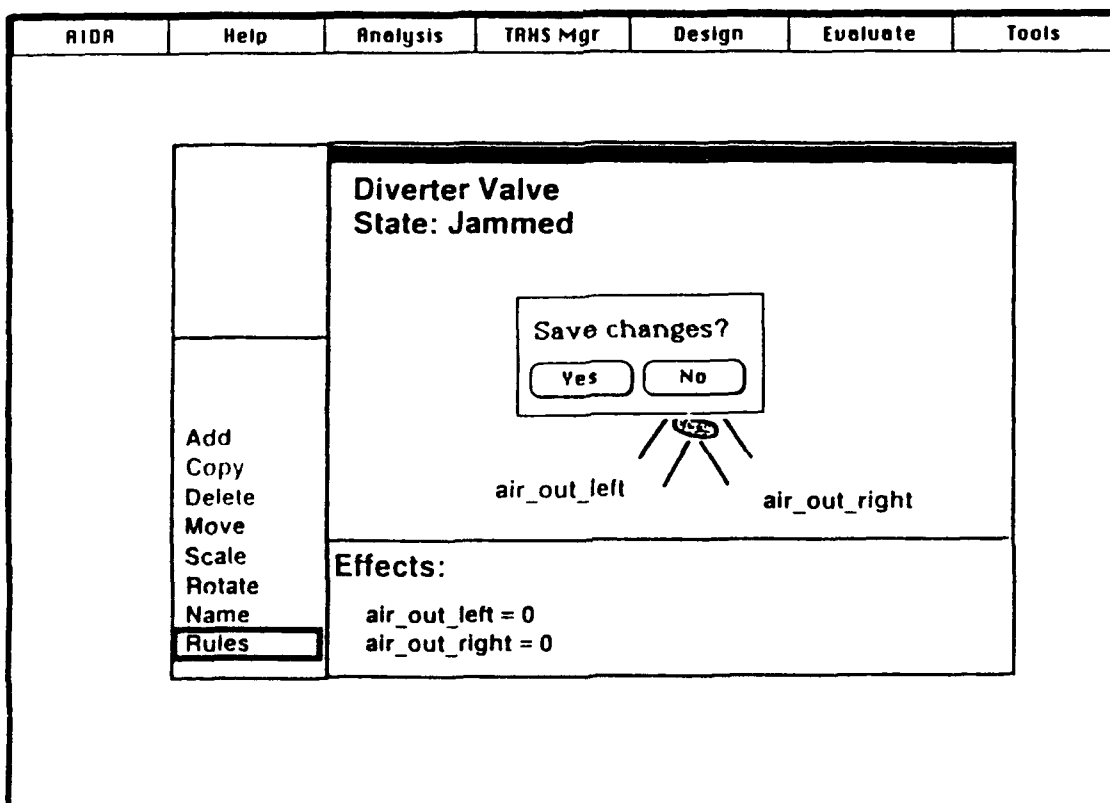
MEI ASSOCIATES INC. # 057. SME completes rules for one state of the Diverter Valve



MEI ASSOCIATES INC. # 058. SME completes rules for one failure mode of the Diverter Valve



MEI ASSOCIATES INC. # 059. SME selects "Quit" from AIDA menu to exit current activity



MEI ASSOCIATES INC. # 060. AIDA determines whether changes are to be saved before quitting

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

Task Analysis

Click on task analysis approach.

Discursive

Graphical

Return

MEI ASSOCIATES INC. #061. SME is asked to select task analysis approach. This time, SME selects "Return" to top level screen

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

TRHS Library

Authoring

Configuring

Detailing

Creating

AIDA

Advanced Instructional Design Advisor

MEI ASSOCIATES INC. #062 SME pulls down "TRHS Mgr" menu and selects "TRHS Library"

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

TRXS Library

Please click button for:

TRHS Index (Alpha)

TRH Classes

TRH Domains

Return

MEI ASSOCIATES INC. #063. SME selects "TRXS Index (Alpha)"

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

TRXS Index (Alpha)

Click on desired TRXS

Analogize	Enterprise	Interpret
Classify/Decide	Evaluate	Judge
Denote	Execute	Propagate
Design	Generalize	Substitute
Discover	Identify	Transfer

Return

MEI ASSOCIATES INC. #064. SME selects Return

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

TRXS Library

Please click button for:

TRHS Index (Alpha)

TRHS Classes

TRHS Domains

Return

MEI ASSOCIATES INC. # 065. SME selects "Return."

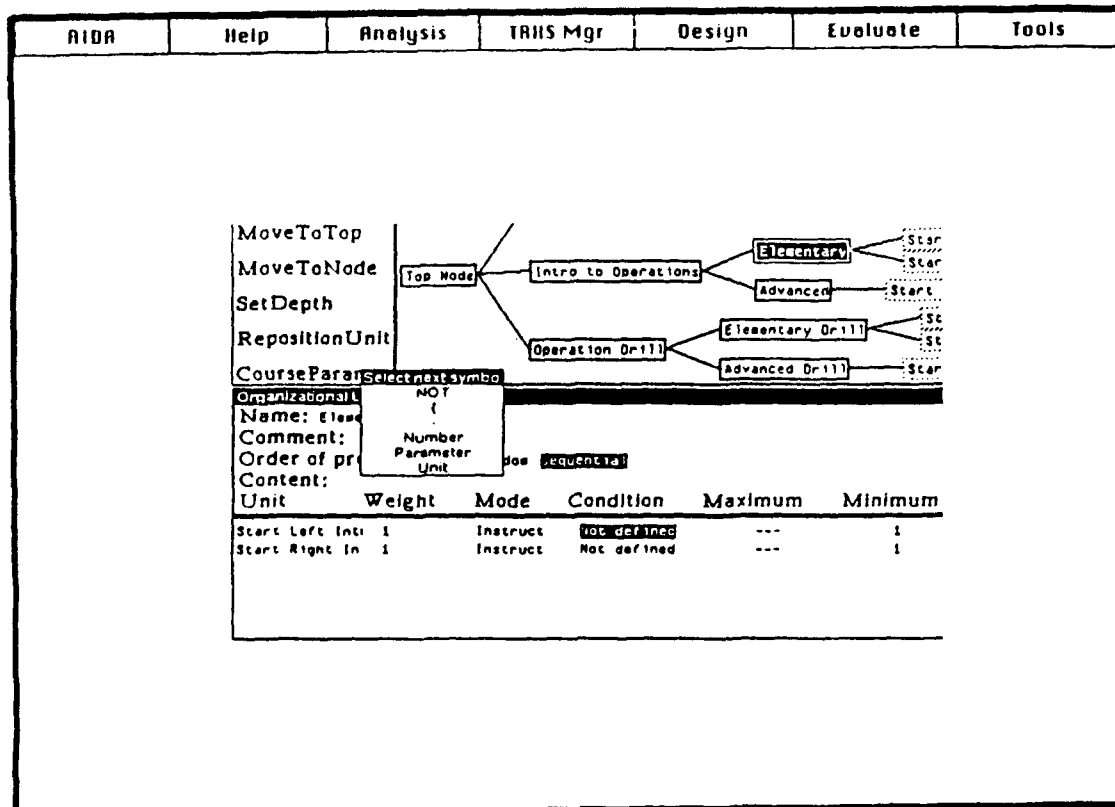
AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
------	------	----------	----------	--------	----------	-------

AIDA

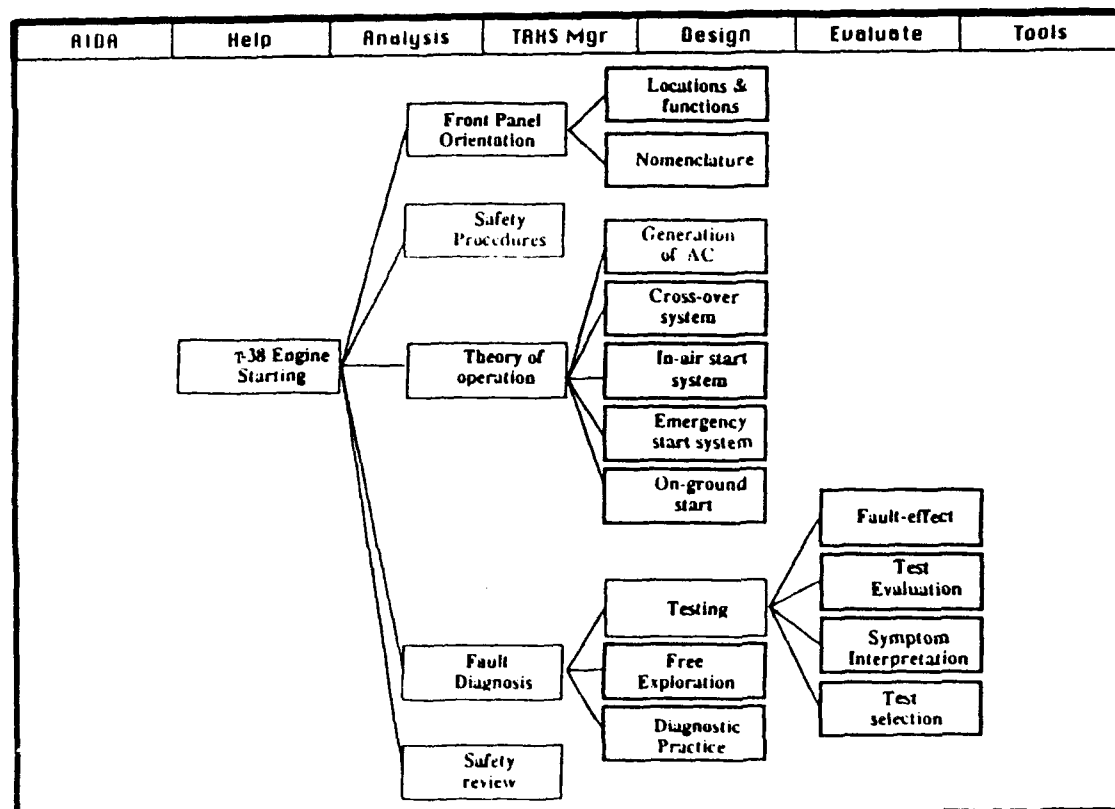
Advanced Instructional Design Advisor

Course
Enterprise TRH

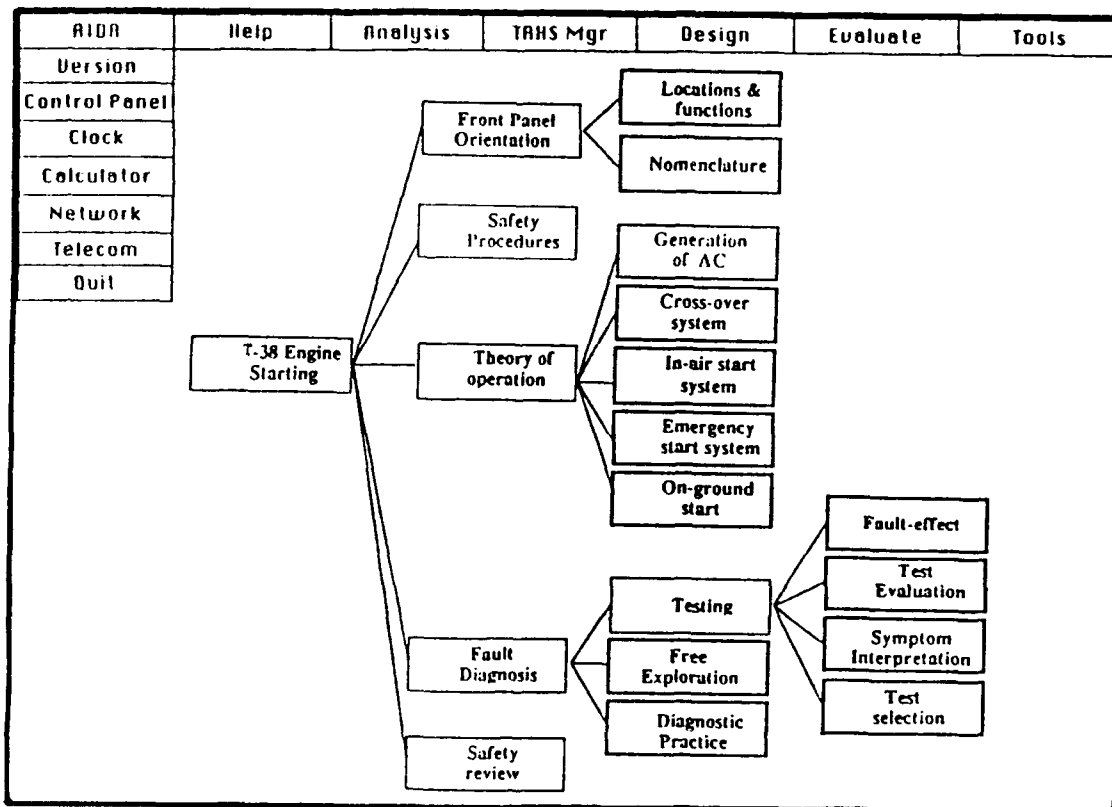
MEI ASSOCIATES INC. # 066 SME pulls down Design menu and selects course



MEI ASSOCIATES INC. • 067. SME begins to author conditional sequence.



MEI ASSOCIATES INC. • 068. SME uses graphical course planning tools



MEI ASSOCIATES INC. # 069. SME pulls down AIDA menu and selects "Quit."

AIDA	Help	Analysis	TRHS Mgr	Design	Evaluate	Tools
<h1>AIDA</h1> <h2>Advanced Instructional Design Advisor</h2> <p>Please enter:</p> <p>Last Name _____</p> <p>First Name _____</p> <p>Middle Initial _____</p> <p>Social Security Number ____-____-____</p>						

MEI ASSOCIATES INC. # 070 AIDA returns to login

APPENDIX C
SYSTEM/SEGMENT SPECIFICATION (DI-CMAN-80008A)

Table of Contents

1.	Scope	39
1.1	Identification	39
1.2	System Overview	39
1.3	Document Overview	39
2.	Applicable Documents	39
2.1	Government Documents	39
2.2	Non-Government Documents	39
2.3	Other Publications	40
3.	System Requirements	41
3.1	Definition	41
3.1.1	Definition of AIDA	41
3.1.2	Definition of XAIDA	43
3.2	Characteristics	43
3.2.1	Performance Characteristics	43
3.2.1.1	Knowledge Acquisition/ Representation (KARS)	43
3.2.1.2	Strategy Analysis	52
3.2.1.3	Transaction Authoring	59
3.2.1.4	Courseware Generation	73
3.2.1.5	Instruction Delivery	73
3.2.1.6	Evaluation	75
3.2.2	System Capability Relationships	75
3.2.2.1	Major Components for AIDA	75
3.2.2.2	Common Capability	76
3.2.3	External Interface Requirements	76
3.2.4	Physical Characteristics	76
3.2.4.1	Protective Coatings	76
3.2.5	System Quality Factors	76
3.2.5.1	Reliability	76
3.2.5.2	Maintainability	77
3.2.5.3	Availability	77
3.2.5.4	Additional Quality Factors	77
3.2.6	Environmental Conditions	77
3.2.7	Transportability	77
3.2.8	Flexibility and Expansion	77
3.2.9	Portability	78
3.3	Design and Construction	78
3.3.1	Materials	78
3.3.2	Electromagnetic Radiation	78
3.3.3	Nameplates and Product Marking	78
3.3.4	Workmanship	78
3.3.5	Interchangeability	78
3.3.6	Safety	78
3.3.7	Human Engineering	78
3.3.8	Nuclear Control	78
3.3.9	System Security	78
3.3.10	Government Furnished Property Usage	79
3.3.11	Computer Resource Reserve Capacity	79
3.4	Documentation	79
3.5	Logistics	80

Table of Contents (Cont'd)

3.6	Personnel and Training	80
3.6.1	Personnel	80
3.6.2	Training	80
3.7	Characteristics of Subordinate Elements	80
3.8	Precedence	80
3.9	Qualification	81
3.10	Standard Sample	81
3.11	Preproduction Sample, Periodic Production Sample, Pilot or Pilot Lot	81
4.	Quality Assurance Provisions	82
4.1	Responsibility for Inspection	82
4.1.1	Contractor's Responsibility for Inspection	82
4.1.2	Government's Responsibility for Inspection	82
4.2	Special Tests and Examinations	82
4.3	Requirements Cross-Reference	82
5.	Preparation for Delivery	82
6.	Notes	83
6.1	Intended Use	83
6.1.1	Missions	83
6.1.2	Threat	83

List of Illustrations

Figure C-1.	Schematic of AIDA Expert System	42
Figure C-2.	Transaction Family for Acquiring an Equipment Mental Model	54

1. SCOPE

1.1 Identification

The system specified in this document shall be identified as the Advanced Instructional Design Advisor (AIDA) built for the United States Air Force Armstrong Laboratory, Human Resources Directorate (ALHRD).

1.2 System Overview

The purpose of the AIDA is to provide intelligent and automated assistance to subject matter experts (SME) and instructional designers (ID) throughout all phases of instructional system development (ISD).

1.3 Document Overview

The purpose of this document is to provide a procurement specification that will satisfy the requirements of the AIDA. The contents of this document defines system requirements which should be accomplished by utilizing commercial Non-Developmental Items (NDIs) for hardware and software components when possible.

2. APPLICABLE DOCUMENTS

2.1 Government Documents

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

- a. MIL-STD-2167A DOD-STD-2167A, February 29, 1988, Defense System Software Development.
- b. MIL-STD-1472C, Human Engineering Design, Criteria for Military Systems Equipment and Facilities, Notice 2, 12 May 1981.
- c. ALHRD, Contract # F33615-88-C-0003, Task Order 0013, Research in Instructional Design in Interactive Technologies.

Copies of specifications, standards, drawings, and publications required by suppliers in connection with specified procurement functions should be obtained from the contracting agency or as directed by the contracting officer.

2.2 Non-Government Documents

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event

of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

1. DESIGN SPECIFICATIONS FOR THE ADVANCED INSTRUCTIONAL DESIGN ADVISOR (AIDA), Final Report, Contract No. F33615-88-C-0003, Task Order No. 0006.
2. DATABASE DESIGN DOCUMENT FOR THE EXPERIMENTAL ADVANCED INSTRUCTIONAL DESIGN ADVISOR (XAIDA), CONTRACT NO. F33615-88-C-0003, TASK ORDER NO. 0013, CDRL SEQUENCE NO. 18.
3. DESIGN SPECIFICATIONS FOR THE ADVANCED INSTRUCTIONAL DESIGN ADVISOR (AIDA), Final Report, Contract No. F33615-88-C-0003, Task Order No. 0013.
4. Mei Associates, Inc., Specifications for an Advanced Instructional Design Advisor (AIDA), 31 July 1990.
5. Mei Associates, Inc., Design Specifications for the Advanced Instructional Design Advisor (AIDA), 24 April 1991.
6. Merrill, M., Li, Z., and Jones, M., Project AIDA: A Concept Paper, Sep 1989.
7. Merrill, M.D., Project AIDA: A theory Paper, Sep 1989.
8. Halff, H.M., Automating Maintenance Training, Jul 1990.
9. Halff, H.M., Teaching Procedures, Nov 1990.
10. Merrill, M.D., ITX for Maintenance Training, Sep 1990.
11. Merrill, M.D. and Jones, M.K., RAPIDS/TRX Walk-through, Oct 1990.
12. Halff, H.M., Automating Instructional Design and Development, Feb 1990.
13. Hickey, A.E., Proceedings of the Working Group Meeting at USU 4-6 Sep 1990, 18 Sep 1990.
14. Hickey, A.E., Proceedings of the Working Group Meeting at San Antonio, TX, 14-16 Jan 1991.

2.3 Other Publications

1. Merrill, M.D. and Li, Z., An instructional design expert system. Journal of Computer-Based Instruction, Summer 1989, v. 15, No. 3, 95-101.
2. Jones, M.K., Li, Z. and Merrill, M.D., Domain knowledge representation for instructional analysis. Educational Technology, Oct 1990.

3. Li, Z. and Merrill, M.D., ID Expert V. 2.0: A desktop instructional design expert system: Part I. Instructional design theory and process, Educational Technology Research & Development, Sep 1989.
4. Merrill, M.D., Li, Z., and Jones, M.K., Instructional transaction theory: Classes of transactions. Utah State University Working Paper, Dec 1990.

Technical society and technical association specifications and standards are generally available for reference from libraries. They are also distributed among technical groups and using Federal Agencies.

3. SYSTEM REQUIREMENTS

3.1 Definition

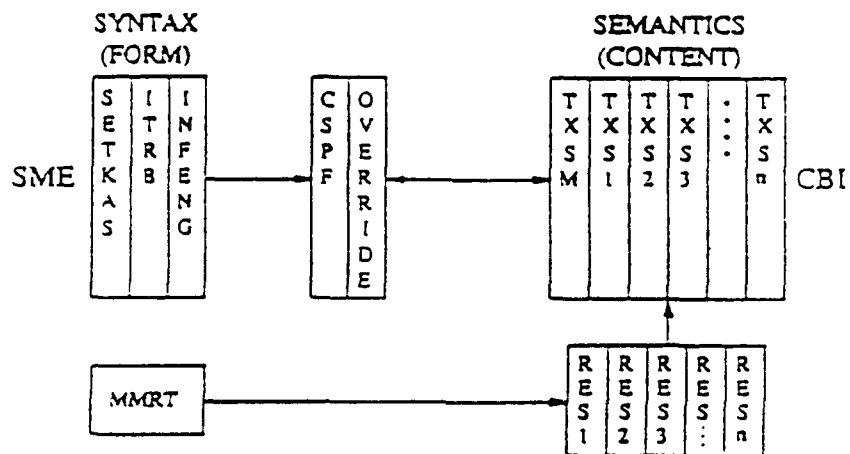
AIDA will be implemented in two stages: the near-term experimental AIDA, called XAIDA, and the long-term fully implemented AIDA. The fully implemented AIDA is briefly defined in Section 3.1.1. The near-term XAIDA, to be implemented over three years in AIDA Phase III, is defined in more detail in Section 3.1.2.

Both AIDA and XAIDA will have six modes of operation, as listed in Section 3.2.1. Some functions, such as task analysis, course generation, and evaluation, will not be fully implemented in the near-term XAIDA. They will be epitomized, however, to ensure that adequate facilities are provided for their future inclusion in the AIDA system.

3.1.1 Definition of AIDA

The AIDA will consist of an expert system as shown in Figure 1, with separate modules for such major components as the knowledge acquisition system, the instructional theory rule base, the inference engine, the transaction shell manager, and the transaction shells themselves. Input materials for the interactive technologies will be prepared by using the facilities of a multi-media resource toolkit.

AIDA will enable the SME/ID to select and implement appropriate presentation strategies for instructional objectives; select and implement appropriate transactions to support those strategies; design screen layouts; develop a lesson prototype based on specified objectives, content, and selected transaction shell; develop a lesson test and conduct a lesson tryout with an editing capability; generate drill-and-practice and tests and grade the results; collect data on instruction usage and student performance.



SME = Subject Matter Expert

SETKAS = Student, Environment, Task Knowledge Acquisition System

ITRB = Instructional Theory Rule Base

INFENG = Inference Engine

CSPF = Course-Specific Parameter File

TXSM = Transaction Shell Manager

TXS1 ... TXSn = Transaction Shells

MMRT = Multi-media Resource Toolkit
(Paint, Draw, Scanner, Audio)

RES1 ... RESn = Resource Files

CBI = Computer Based Instruction

Not Shown: Help System
Audit Trail

Figure C-1. Schematic of AIDA expert system.

3.1.2 Definition of XAIDA

XAIDA will be designed with basic information about the students and the instructional environment already in the system. The instructional designer will be given the learning capabilities to enter. XAIDA will then select and configure transaction shells appropriate to the specified capabilities. The instructional designer will then be prompted to enter any needed content knowledge to complete a frame appropriate to the particular knowledge type.

Because we are assuming a fixed instructional environment (small class, computer-based instruction, located at a TTC) and students who are readers and reasonably motivated, information about the students and environment will be hard-coded into the XAIDA EXECUTIVE for the time being. Because we are focusing on teaching procedures for avionics maintenance training, we will customize an enterprise analysis pertinent to that domain and also customize an elaborated frame network shell pertinent to electronics maintenance training procedures. Because these are shells (and can also serve as variables in AIDA experimentation), they are retained as place-holders in the XAIDA EXECUTIVE pending further development of the system.

3.2 Characteristics

3.2.1 Performance Characteristics

The AIDA system will have six modes of operation:

1. Knowledge Acquisition/Representation (KARS)
2. Strategy Analysis
3. Transaction Authoring
4. Courseware Generation
5. Instruction Delivery
6. Evaluation

3.2.1.1 Knowledge Acquisition/Representation (KARS)

Knowledge analysis is the step whereby knowledge of the domain to be instructed (in this case, avionics maintenance procedures) is elicited from subject matter experts (SME) and represented in a domain knowledge base which may be used by all other steps and tools in the course development process. It is accomplished by using the Knowledge Acquisition/Representation System (KARS) described in Section 3.2.1.1. The System/Segment Specification for the XAIDA KARS will be developed in Year 1 of AIDA Phase 3 for implementation in Year 2. AIDA KARS may be modified in future years to accept input from a Cognitive Task Analysis (CTA) tool, such as that under development by ALHRD/MO. A CTA system is described in Section 3.2.1.1.

The Knowledge Representation Model. Knowledge is represented by objects called frames; each frame has an internal structure and external links to other frames. These external links are termed

elaborations of the frame. A set of elaborated frames linked together, containing all the knowledge required for instruction leading to acquisition of an integrated human performance, or enterprise, is called an elaborated frame network.

There are three kinds of frames:

- o entities, corresponding to some thing, for example a device, object, person, creature, place, or symbol;
- o activities, groups of related actions to be performed by the learner; and
- o processes, groups of related actions entirely external to the learner.

There are four kinds of elaborations. These are:

- o attributes, which represent characteristics of a frame;
- o components, which represent constituents of a frame For an entity, the components would be parts of the entity; for an activity, steps; and for a process, events and causes;
- o abstractions, which correspond to a "kinds-of" class/sub-class hierarchy into which the frame may be classified;
- o associations, links to other frames in the network.

The network structure of the knowledge representation allows information to move through the structure, so that data contained in one part of the net affects the data stored elsewhere. Two principal means by which this occurs are:

- o inheritance, in which attributes of a class or super-class in an abstraction hierarchy are passed to a sub-class or instance;
- o propagation, in which the contents of a frame influence the contents of another frame connected to it via an association link.

Entity Frames. Entities are things in the real or imagined world including objects, creatures, places, and symbols. At least one entity must be involved.

Activity Frames. An activity is some group of actions performed, or which could be performed, by the student.

Process Frames. A process is some group of actions outside the student, including physical and social events in the real or imagined world.

Relations. In addition to the frame, the other fundamental structure is the relation. Relations are structures that link and attach meaning to a set of frames. Each frame links to the relation, and from the relation to other frames in the set, in a manner specified by the relation. The semantics of the particular relation give meaning to the individual links.

Elaboration, and Elaborated Frame Networks. Frames have an external structure of links to other frames. These are termed elaborations of the frame. Adding relations (links) to a frame is called elaborating a frame. A set of linked elaborated frames is termed an elaborated frame network, or EFN. A single EFN corresponds to the knowledge elements and interrelations required to support performance of an enterprise. A course may facilitate acquisition of one, or several, enterprises, thus the domain knowledge base for a course may contain one or a set of EFN. A set of EFN is itself an EFN.

There are four kinds of elaborations.

- o attributes, which represent characteristics of a frame;
- o components, which are the constituents of a frame;
- o abstraction, which represents the generality of frames;
- o association, non-hierarchical aggregations of frames.

The Attributes Elaboration. An attribute is a labeled set of values from which objects and their properties may take values over time. Attributes define the characteristics of frames. The constraints placed on a particular attribute determine the legal values that may be taken by objects possessing that attribute. The following operators may be used to define legal values for attributes: the booleans AND, OR, XOR, NOT; the logicals =, <>, <, <=, >, >=; and the range operator .. . Attributes take values from the set of legal values defined for the attribute. The value selected during analysis is termed the "initial" value. In addition, storage may also be reserved for a current value.

The Components Elaboration. Each kind of frame has its unique component structure: (1) Entities have parts, (2) activities have steps, (3) processes have stages.

(1) Entity Components. An entity can be described by its parts. Each part has at least a name and an associated function. If the entity is a physical object, then each of its parts may also have a location and a graphical description. A part, of course, can have sub-parts.

(2) Activity Components. An activity consists of one or a series of steps. Steps are performed in a specified sequence, including loops and conditions. Each step has a set of associated actions. These actions may be performed in a specified sequence (algorithm), including loops and conditions, or they may be triggered by events (heuristics). All actions are represented in the analysis as action + trigger(s) + consequence(s). Triggers and consequences are either sequence data, changes in attributes,

time values, or a combination. Each activity must link to at least one entity which is an actor (performs action and is capable of varying its behavior in the activity based on some internalized knowledge). A step, or an action, of an activity may itself be an activity, with its own sub-steps.

(3) Process Components. The constituents of a process are its stages; each stage has an associated event topology. An event may itself be a process, or in this context, a sub-process. An event is a transformation of inputs to outputs. Inputs are either attributes, actions, or time values. Outputs are either attributes, actions, time values, or stage transitions. Transformations are either mathematical, logical, or both. An event may be encapsulated within an entity, or be abstract. Events may be linked together into event topologies. A topology is like a network, except that it doesn't need to be fully interconnected. The network structure supports dependencies among events (the output of event A is the input to event B), feedback (the output of event B becomes an input back into event A), and self-regulation. The addition of timing signals as inputs and outputs (these may be relative or absolute) supports synchronization and temporal dependencies. An event topology describes how a process behaves in respect to changes in its inputs. A stage is a sequentiable phase within a process which has an event topology that is distinct from those of other stages. A process need only have one stage.

The Abstraction Elaboration. Abstraction represents the generality of entities, activities, and processes. The levels of abstraction are instance and class, with relationships among classes in a multi-level generalization hierarchy represented by sub-class and super-class links. The abstraction elaboration is exactly equivalent to generalization. An instance represents a specific entity such as a particular object, person, symbol, or place. It may also represent a particular activity, or a particular process. The instance represents the lowest level of abstraction and has no subordinates. A class is an abstract frame that represents general features held in common by two or more frames. Attributes, and their legal values, help define the class. Instances in the class share these attributes.

The Association Elaboration. Associations are non-hierarchical aggregations of frames. For an association, each frame in the relation may be thought of as an aggregate of the other frames.

The Collection Elaboration. Collections are sets of frames all of the same class. Unlike the other relations, collections are not considered to be elaborations. A collection may be substituted for frames in certain situations. Collections, however, do not have the same status as frames; rather it is the objects of which a collection is formed that are the frames. In semantic data model terms, a collection is a grouping relation. There can be no inheritance from a collection to its members. Generalization and inheritance are defined on the class structure

of the underlying frame, not the collection. Collections, however, may have attributes associated with them. These non-inheritable attributes perform functions such as summarizing across the collection.

The relationship between a collection and its underlying frames is extended to allow the definition of collections of collections. Collections may take the place of frames in associations. Collections may also take the place of frames as attributes, and as components. A single frame may be a member of more than one collection. Collection membership is defined by an expression using a syntax equivalent to that for defining class membership.

---***---

Knowledge Acquisition. A method is required for the identification of entity and process components. One method is simulation. In the steps of building the device simulation, production of objects identifies entities and their components, and the specification of behavior of objects identifies the event topologies of processes. The method also generates parts of the EFN knowledge structure automatically, and is augmented by other methods to acquire other elements of the subject matter, such as activity components, and abstraction. XAIDA will employ a device simulation system.

Cognitive Task Analysis. The first step in the design of any instruction is a task analysis to determine what should be taught. A behavioral analysis is not sufficient; a cognitive analysis is required to take into account the cognitive processes involved in learning and performance.

Half (1990) identified three types of cognitive structures important to the maintenance enterprise: (1) the execution of procedures, (2) a mental model of the equipment, and (3) fault isolation skills. Thus, an adequate cognitive task analysis should identify the information and skills that must be imparted to the student to support the acquisition of these cognitive structures. XAIDA will address only the areas of procedures and mental models.

Cognitive Analysis of Procedures. Kieras advocates a GOMS (Goals, Operators, Methods, and Selection Rules) analysis in which the tasks to be accomplished are broken down into a series of goals and subgoals, recursively, until a level is reached in which the subgoal can be achieved by either a primitive level motor or mental act.

(1) Goals represent a person's intention to perform a task, subtask, or single cognitive or physical operation. They are organized into structures of interrelated goals that sequence methods and operations.

(2) Operations are elementary physical actions (e.g., pressing a button) and cognitive or mental operations (e.g., reading a voltage and storing it in working memory). The most primitive mental operations are actions such as receiving perceptual information, making a basic decision, depositing facts from working memory into long term memory, retrieving facts from long term memory and activating them in working memory, forming a goal, etc.

(3) Methods generate sequences of operations that accomplish specific goals or subgoals. The goal structure of a method defines its internal organization and control structure. The GOMS model assumes that execution of a task or procedure involves decomposition of the task into a series of subtasks. A skilled person executing a procedure has effective methods for each subtask. Accomplishing a task involves executing the series of specialized methods that perform each subtask. A person's knowledge of how to do a complex task is a mixture of high-level methods, (i.e., task-specific information, and low-level methods, i.e., system-specific knowledge.

(4) Selection rules determine which method to select. The selection rule must state the appropriate context for using any given method. If there is more than one method, the rule must state when each method is appropriate.

In summary, the GOMS model characterizes the user's knowledge as a collection of hierarchically organized methods and associated goal structures that sequence methods and operations. The knowledge captured in the GOMS representation describes both general knowledge of how the task is to be decomposed and specific information on how to execute the methods required to complete the task.

Kieras has prepared a detailed guide for doing task analysis of procedures using the GOMS methodology (Kieras, 1988a). He has also defined a language call (NGOMSL) or "Natural" GOMS Language which is relatively easy to read and write. Kieras' guide also includes procedures for doing a GOMS analysis by using a breadth-first expansion of methods rather than trying to describe goal structures directly.

Mental Models Analysis. Halff (1990) summarized the importance for maintenance training of imparting correct and adequate mental models of the equipment. Kieras (1988b, 1990) pointed out that the most accurate way of determining the mental model to be taught would be to do a complete cognitive simulation. However, realizing that this is not always a feasible approach, he spelled out some heuristics that could be used to determine the mental model that should be taught in lieu of a complete simulation. The heuristics are:

- relevance to task goals
- accessibility to use
- critical procedures and inference strategies

These heuristics involve doing a GOMS-like analysis. In addition two other hierarchical cognitive analyses are required: an explanation hierarchy and a hierarchical decomposition of the device structure and mechanisms.

The "relevance to task goals" heuristic explanations should be given only if they are relevant to a task goal. To carry out this heuristic, an explanation hierarchy is constructed. The first pass at what goes into this hierarchy can be what is in the existing documentation. The goals of the GOMS analysis are then mapped to the explanation hierarchy, which will reveal any missing explanatory information as well as any extraneous material which need not be taught. Constructing the explanation hierarchy is not really extraneous work since this material is needed for the instructional material. For instance this is the material that goes into the message windows in the RAPIDS system.

The second heuristic, "accessibility to use", implies that the device illustration or simulation which is presented to the technician should not contain parts which cannot be accessed. Again this involves mapping the GOMS analysis, but onto the device description, rather than the explanation hierarchy.

The third heuristic says that the GOMS analysis should be examined for procedures that will be difficult to learn due to what appears to be arbitrary content. These procedures should then be analyzed to determine what inferences would need to be made in order for the content to appear logical rather than arbitrary. The information necessary to make those inferences should then be made explicit in the training materials. This information will need to be included either in the explanation hierarchy or the device description.

Level of Detail of the Task Analysis. One way to determine the content of instructional materials and training procedures is to do a complete cognitive simulation of a given task. The advantage of a simulation is that it insures that the analysis is complete. A tremendous amount of information about the task at hand can be gained if the analysis is completed down to the level of simple operations or operators for most aspects of the task. The information that can be derived from the simulation includes the time to learn the task, the amount of transfer of training from one procedure to another, and the execution time for various procedures or methods. The GOMS method does not require that it be followed through by a complete simulation or that all tasks be analyzed to the level of simple operators.

How low the level of analysis needs to be for the procedures for any given instructional package, will be determined in large part by the level of expertise of the trainees. High level tasks may

need to be broken down into subtasks. Anderson refers to this as adjusting the grain level of the instruction.

To decrease the workload of authoring the simulation and/or doing the GOMS analysis for a given domain, AIDA will contain a library of generic low-level procedures, such as testing igniter plugs. These modules can also be given as screening tests to insure that these low-level methods are learned before entering simulations at a higher level or aimed at specific problems.

Representing The Task Analysis. The results of a NGOMSL analysis will be implemented in a working simulation. The device knowledge necessary to carry out the simulation will be represented in a schema such as Anderson's PUPS (Penultimate Production Systems). PUPS theory, which holds that procedures are acquired by compiling declarative knowledge, is compatible with Merrill's Transaction Shell representation. The declarative knowledge necessary for compiling the procedures which model the task performance is represented in schema-based structures called PUPS structures. These schema include slots for the function of the entity being represented by the schema, a form slot for the physical appearance of the entity, and a precondition slot which states the preconditions necessary for the function to be achieved. In compiling the productions which are the basis of procedural knowledge, the function slot maps to the goal to be achieved which will require knowledge of the entity represented; the preconditions slot maps onto the condition of the condition-action pair in a production. The form slot in the PUPS system holds the form of the current action to be carried out such as a particular LISP function. A similar scheme could be used for representing the GOMS analysis. Merrill has proposed an activity frame that has paths or sequences of actions. This frame could also have slots for the function, the operators, and the outcome. The values for these slots could probably be automatically generated from a NGOMSL analysis just as it is technically feasible to generate a running production rule-based simulation from a NGOMSL analysis.

The representation scheme proposed by Merrill for AIDA will be used to represent the explanation hierarchy. The device knowledge will ultimately be represented in the graphical simulation. The initial representation may be a hierarchical listing of the names of the device components or a block diagram, which can serve as a guide for constructing the sketch which in turn will guide the construction of the graphical simulation.

Mapping the Content of GOMS and Mental Model Analysis to the Device Simulation. Kieras' approach yields three hierarchically arranged representations. First, the GOMS analysis spells out the steps to be followed in carrying out procedures for operating, calibrating, troubleshooting, or repairing the equipment starting with the highest level goals and methods. These are successively decomposed to lower-level subgoals and methods. Second, the GOMS analysis identifies any device

components that need to be included in the representation of the device structure as well as the declarative knowledge that needs to be conveyed about them: function, location, name, etc. Third, the explanation hierarchy contains the causal and declarative knowledge necessary to execute the procedures, support inferences necessary for constructing a mental model of the equipment, and define the attributes and rules of objects, etc.

The device simulation in a system such as RAPIDS contains a graphic representation of the device structure and qualitative simulations of its functioning. Authoring in the RAPIDS II simulation starts with a temporary sketch which is derived from the prior cognitive analysis, particularly the mental model analysis which entails inter-relating the GOMS analysis, the explanation hierarchy and the hierarchical device structure decomposition. However, the construction of the simulation is done in bottom-up fashion starting with the lowest level of the device hierarchy. The lowest level objects are the bottom items in the device structure analysis, corresponding to the objects manipulated by the lowest level operators in the GOMS model. For this reason, it is not feasible to develop the simulation and do the GOMS analysis and explanation hierarchy in parallel. The analyses have to be complete before construction of the simulation can begin.

The behavior of the objects are defined by attribute handles and rules. These aspects of the simulation are drawn from the explanation hierarchy. Once the basic simulation is complete, procedures which are carried out on the device are authored by carrying out a sequence of actions which correspond to actions spelled out to accomplish the goals in the GOMS analysis. The individual actions correspond to the operators. What is missing from the simulation representation is any indication of the function or purpose, i.e. goals, of the procedure. These have to be represented in the dialogue windows.

Aids for Doing a Cognitive Analysis. At some future date, a task analysis approach similar to that developed by Kieras will be added to AIDA KARS. This includes (1) a GOMS analysis for the procedural aspects of the task, (2) a mental model analysis to develop an explanation hierarchy, and (3) a decomposition of device structure and function and relating them to the GOMS analysis. Shells will aid in the analyses.

The GOMS shell will guide the novice in doing a GOMS analysis of a particular task using either the documentation at hand or the knowledge of a subject matter expert. The shell will be based on Kieras' manual on how to do GOMS analysis and uses his English-like language for representing the analysis. Kieras' guide contains many rules of thumb which can be implemented in a knowledge-based shell to give guidance to the SME or ID. For instance, Kieras recommends that a given method contain no more than five steps. If there is more than five, some may need to be grouped into a higher-level method. There is also guidance on

creating generic methods to represent methods which occur often in slightly different context.

The GOMS shell will do much of the bookkeeping necessary for a GOMS analysis, such as creating a list of methods and information identified by the methods that need either already to be known or taught, such as their location, etc. A more sophisticated shell could automatically map the results of the analysis into the knowledge representation system (KARS). A less sophisticated system would create a paper guide for what should be hand entered into KARS. Similar shells will also be created for the explanation hierarchy and the device structure and function knowledge.

3.2.1.2 Strategy Analysis

Strategy Analysis identifies the enterprises to be learned, and selects and sequences transactions to instruct the enterprises. Additionally, information about the audience and the instructional setting are gathered in this phase.

Enterprise Transactions. An enterprise is a complex human performance that requires an integrated set of knowledge and skills. The goal of instruction is the acquisition by the learner of one or more enterprises. The primary transaction shells, described in Section 3.2.1.3, facilitate the acquisition of the knowledge and skills which comprise enterprises, but by themselves cannot accomplish their integration. This integration must be accomplished by transactions at the enterprise level.

Transaction Families. The transactions necessary to acquire all the knowledge and skill associated with a given enterprise comprise a transaction family. In maintenance, the six enterprises are equipment operation, equipment calibration and adjustment, equipment testing, access and disassembly, equipment repair, and troubleshooting. The type and sequence of interactions necessary to acquire each of these complex enterprises is different.

Transaction Manager (TXSM). A high level transaction manager is required for each of the enterprises. The transaction manager is a program that calls and sequences the primary transactions identified as necessary for this curriculum.

The Equipment Model TRX Family. In addition to transaction families for each of the 6 maintenance training enterprises, an equipment model family of transactions is a component of each of the other transaction families. Figure C-2 identifies the 5 transaction shells required to construct a mental model of a particular equipment. The equipment model transaction family is not a "stand-alone" family, but is a component of the transaction family required for each of the other 6 maintenance training enterprises. Two classes of transactions are represented in the equipment model:

identify - (1) physical & conceptual structure; and interpret - (2) device functioning, (3) device configuration, (4) fault recognition, and (5) prediction. It also provides for an integration of the learning facilitated by the primary transactions, ideally as a performance or simulation of an activity that is representative of the real-world performance of the enterprise.

A course is a set of enterprise transactions and their supporting families of primary transactions. A course organization is a nesting and/or ordering of the enterprise transactions.

Classes of Enterprise Transactions. There are six classes of enterprises: denote, evaluate, execute, design, interpret, and discover.

(1) A Denote enterprise transaction requires as a focus one of the following: a primary transaction from the Component class - either an Identify transaction for an entity, an Execute transaction at the Denote level of performance for an activity, or an Interpret transaction at the Denote level of performance for a process frame - or a Classify/Decide primary transaction from the Abstraction class. (Performance level is a parameter to Execute and Interpret transaction shells. For Execute, the values may be either Denote or Perform. For Interpret, values are either Denote, Explain, or Predict.) Primary transactions from the component, abstraction, and association classes are included to support the focus transaction. Performance for a denote enterprise is characterized as knowing about something.

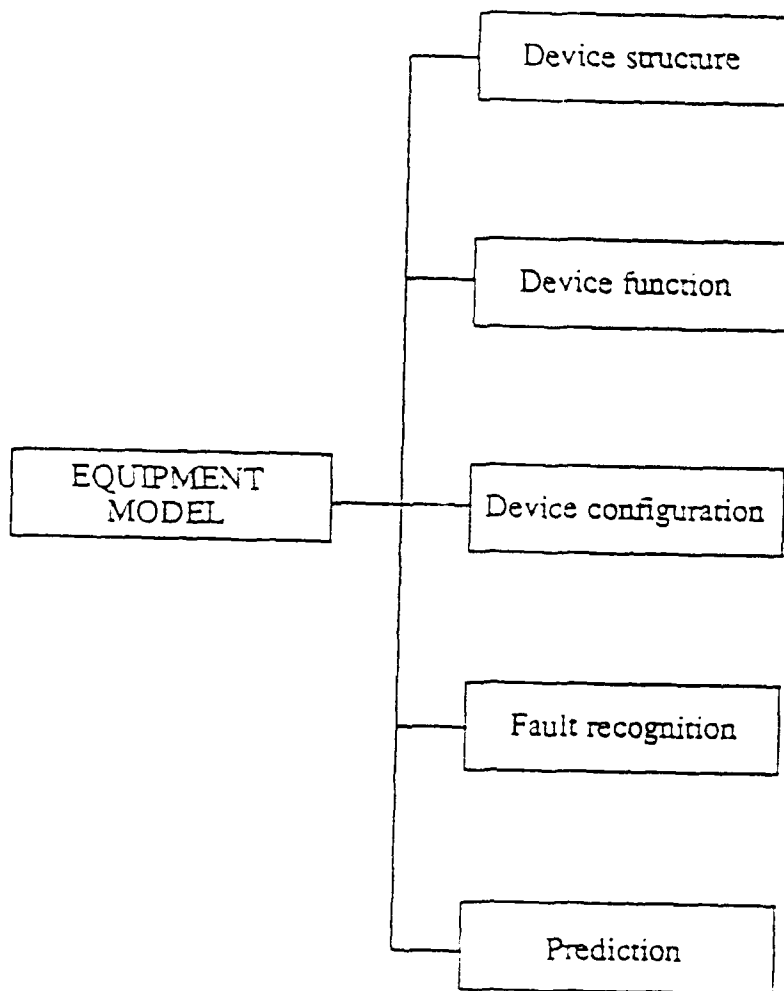


Figure C-2. Transaction family for acquiring an equipment mental model.

With a component class primary transaction as focus, the enterprise transaction enables the student to describe the parts, their functions and locations for an entity; describe the steps for an activity; or describe the events for a process. With a Classify/Decide primary transaction as focus, the enterprise transaction enables the student to identify instances or discriminate kinds.

(2) An Evaluate enterprise transaction requires as a focus a Judge primary transaction from the Abstraction class. The Judge transaction instructs an abstraction hierarchy of entities, activities, or processes. Primary transactions from the component, abstraction, and association classes are included to support the focus transaction. Performance for an Evaluate enterprise is characterized as classifying and ranking the adequacy of an entity, the performance of an activity, or the effectiveness of a process.

(3) An Execute enterprise transaction requires as a focus an Execute primary transaction (at the Perform level) from the Component class. The content for the focus transaction is the steps of an activity frame. Primary transactions from the component, abstraction, and association classes support the focus transaction. Performance for an Execute enterprise is characterized as performing some activity.

(4) A Design enterprise transaction requires as a focus a Design primary transaction from the Association class. Primary transactions from the component, abstraction, and association classes support the focus transaction. Performance for a Design enterprise is characterized as inventing or creating a new artifact. It enables the student to design a new entity or activity not previously instructed.

(5) An Interpret enterprise transaction requires as a focus an Interpret primary transaction, at the Explain or Predict level of performance, from the Component class. The content for the focus transaction is the events and causal network of a process frame. Primary transactions from the component, abstraction, and association classes support the focus transaction. Performance for an Interpret enterprise is characterized as knowing why some process works.

(6) A Discover enterprise transaction requires as a focus a Discover primary transaction from the Association class. Primary transactions from the component, abstraction, and association classes support the focus transaction. Performance for a Discover enterprise is characterized as finding a new relationship or process. It enables the student to discover a new entity or process not previously instructed.

Authoring Enterprise Transactions. The enterprise transaction is authored by selecting a focus transaction and sequencing it with the supporting primary transactions. Course

organization consists of sequencing the enterprise transactions themselves, plus the sequencing of primary transactions within each enterprise transaction.

Sequencing Alternatives. There are two dimensions of sequencing at the enterprise level, yielding seven sequencing alternatives or approaches. They are the (1) primary sequence, and (2) secondary sequence.

(1) Primary Sequence, the first dimension, includes Encyclopedic, Case Study, and Situational sequences.

(a) The Encyclopedic sequence systematically calls each primary transaction to instruct elements of the content, eventually integrating these at the enterprise level. This type of sequencing is often found in textbooks and reference manuals.

(b) The Case study sequence presents a sequence of examples, scenarios, or cases of the focus transaction and the necessary supporting transactions, with each case being complete in and of itself. The sequence of cases is graded on some dimension, such as familiarity, frequency, or criticality.

(c) The Situational sequence is characterized as on-the-job learning, where instruction is delivered on an as-needed basis. Only that instruction necessary to the immediate task is presented; integration must occur opportunistically. Situational sequence is facilitated by an online advisor system and student modeling.

(2) Secondary Sequence, the second dimension, includes Elaboration, Prerequisite, and Flat sequences.

(a) The Elaboration sequence starts with a simple, representative element or elements of the focus content, and progressively adds layers of detail as the instruction progresses.

(b) The Prerequisite sequence orders elements of subject matter based on their dependency interrelations. The focus content is at the top level of the hierarchy.

(c) The Flat sequence involves no systematic ordering at the secondary level.

Approaches to Sequencing: The primary and secondary sequences described above may be combined into seven approaches to sequencing: elaborated, prerequisite, and flat case study; elaborated, prerequisite, and flat encyclopedic; and situational.

(1) The Elaborated case study requires a number of cases of the focus transaction, each complete in of it itself.

(2) The Prerequisite case study selects cases equivalently, but the secondary sequence follows a prerequisite hierarchy. The case is overviewed in the focus transaction, but then instruction is built bottom-up following the prerequisites. Each supporting transaction may be called up one or more times at different nodes in the hierarchy. The next case is handled in a similar way, refreshing and reviewing content that had already been introduced in earlier cases, and introducing additional content.

(3) The Flat case study has no systematic secondary ordering. Once a case has been selected, instruction begins with an overview from the focus, then each supporting transaction is called in turn to present all required content for that case, finally returning to the focus for a full presentation. Then the next case is selected.

The encyclopedic sequences are not built on cases. Any abstraction hierarchy is taught as part of the supporting content, rather than being used to generate cases.

(4) The Elaborated encyclopedic sequence begins with a representative element or elements of the focus content, introduces supporting content as needed, then builds to the full focus content.

(5) The Prerequisite encyclopedic sequence begins with an overview of the focus content, then goes to the lowest levels of a prerequisite hierarchy and sequences the primary transactions to deliver instruction for nodes on the hierarchy, building eventually to a full focus transaction.

(6) The Flat encyclopedic sequence begins with an overview of the focus, then each supporting transaction is called in turn to present all required supporting content, finally returning to the focus for a full presentation.

(7) The Situational sequence delivers instructional elements on demand, either as a result of a student request or based on an online determination by an advisor program of the learning requirements of the student.

Making Sequence Decisions. Authoring the sequence for an enterprise transaction involves the following steps:

- 1 determine enterprise content;
- 2 select the focus transaction;
- 3 for each content grouping, select supporting transactions;
- 4 select primary and secondary sequence;
- 5 if case study, create instances for the focus content;

- 6 identify content for each case;
- 7 if prerequisite sequence, identify prerequisite relations;
- 8 if elaboration sequence, identify elaboration levels; and
- 9 configure parameters for each call to a transaction.

Step 1, Determine Enterprise Content, begins when the author selects the focus content, such as an activity or process frame. The system then initiates a spreading activation search of the EFN, following relations from that frame. For each relation leading from the frame, the author indicates whether that relation should be included in the enterprise. The search continues from the related frames for any relation that is included, while a path is terminated for any relation not included. The search continues until all paths have either reached their end or been terminated. The system then creates the enterprise transaction structure, and stores a representation of the subset of the EFN that has been selected as the content for the enterprise.

Step 2, Select the Focus Transaction, is performed by the author based on a recommendation by the system. The recommendation is based upon the transaction class appropriate for the content structure of the focus.

Step 3, Select Supporting Transactions for each content grouping, is performed by the author with consultation from the system. The system parses the content into content groupings according to the classes of primary transactions (component and abstraction hierarchies, and association links). Each grouping is presented in turn to the author, along with recommended transactions for that grouping. Recommendations are based first on the transaction class appropriate for the content grouping, and may be further refined by environmental parameters, such as the availability of specific resources. Recommendations may also take into account compatibility with the focus transaction. For example, if the selected focus transaction uses video, employs a particular instructional technique, or is optimized for a given domain area, then the recommended supporting transactions will take these conditions into account. Recommendations are ranked if there is more than one possible choice.

Step 4, Select Primary and Secondary Sequence alternatives for the enterprise transaction, is performed by the author based on recommendation of the system.

Step 5, if Case Study, creates instances for the focus content. The author selects an abstraction hierarchy from which cases will be drawn and identifies the attribute which will be used to order cases. The system then prompts the author to return to

knowledge analysis to identify the instances for the classes in the hierarchy which will form the cases.

Step 6, Identify Content for Each Case, is performed automatically by the system parsing the subset of the EFN selected in step 1, and selecting any content related to the focus or the instance.

Step 7, if Prerequisite Sequence, Identify Prerequisite Relations, is performed by the author using a system tool to identify dependency relations. This relation structure is stored with the enterprise. If prerequisite case study, the prerequisite relations for each case may be derived automatically from this structure.

Step 8, if Elaboration Sequence, Identify Elaboration Levels, is performed by the author. The number of levels, and the content for each level of elaboration, is identified. If case study, this is performed for each case. This data is stored with the enterprise transaction. Secondary content for each level of elaboration is sequenced by the prerequisite relations, if available, or flat. At this point, all primary and secondary sequencing has been completed.

Step 9, Configure Parameters for each call to each transaction, is performed by the author. The system brings up the configuration for each call in turn, and presets as many parameters as possible. These include the content for the call, based on the earlier steps, and the values of other parameters based on either student attributes and/or earlier configuration decisions for that enterprise. In addition to the normal configuration capabilities, the author may set a parameter for all calls to the transaction from this enterprise, for all calls to any transaction having the parameter from this enterprise, or for all calls to any transaction having the parameter from this course. (An example of the latter might be setting display or response parameters to establish a uniform interface across the course.)

3.2.1.3 Transaction Authoring

An instructional transaction is a particular instructional interaction with a student; it is a bounded interchange between an instructional system and a student which facilitates the acquisition of a specified competence in the student. Transactions comprehend the entire range of instructional interactions including: one-way transmission of information (e.g. video, lecture, or document); discussions and conversations; leaving aside ITS and microworld, but not simulations as used in non-ITS settings; simulations; and micro-worlds (with or without coaching).

XAIDA will provide a library of reusable instructional programs, or transaction shells, for the delivery of instruction. These

programs will contain generalized instructional algorithms, each appropriate for teaching a certain type of content, but will not contain any content. Each shell will incorporate a number of parameters, configurable by the author, which control the functioning of the shell during course delivery.

A transaction can assume both expository and inquisitory modes; it allows the degree of learner or system control to be adjusted; it includes display and response parameters which allow the transaction to be customized for different learners, different subject matters and different delivery systems. Each transaction can perform different instructional functions with its content: overview; familiarity instruction; basic instruction; example; practice; remediation; and assessment.

Transaction Shells A transaction shell is a piece of computer code which when executed causes a given transaction to take place. A transaction shell knows what knowledge it must have in order to execute its interaction with the learner. It is able to query the domain knowledge base to find the required knowledge and thus be able to instantiate its knowledge slots. If the domain knowledge base does not contain the necessary knowledge the transaction shell can direct the SME/ID to supply the required content. Once a transaction has been selected or prescribed, it must then be configured and authored.

Configuration involves setting the parameters, modifying the strategy, and attaching the content. Authoring involves attaching domain specific instructional materials to the instructional structure set up by the transaction. Each transaction shell has default values for each of its parameters, including its strategy elements. Transaction shells reside in a transaction library. Configured and authored components are also stored in the library.

Classes of Transaction Shells. AIDA recognizes four classes of transactions, with each class differentiated from the others in terms of the knowledge structures and performance components instructed. The four transaction classes are component, abstraction, association, and enterprise. Component transactions instruct all or part of one component hierarchy (parts, steps, or events) in the elaborated frame network. Abstraction transactions instruct all or part of an abstraction hierarchy. Association transactions instruct two or more frames linked by an association relation. Within each class are a number of subclasses. There are 12 subclasses: For components, the subclasses are identify, execute, and interpret; for abstraction, the subclasses are judge, classify/decide, generalize, and transfer; for association, propagate, analogize, and substitute.

Component Transactions. There are three subclasses of component transactions corresponding to the three types of knowledge

frames: identify for entity frames, execute for activity frames, and interpret for process frames.

(1) An identify (what) transaction requires either an instance or class entity frame. It enables the student to acquire the names, functions, properties, and relative location of all the parts which comprise an entity.

(2) An execute (how) transaction requires either an instance or class activity frame. It enables the student to acquire the steps of the activity.

(3) An interpret (why) transaction requires either an instance or class process frame. It enables the student to acquire the stages in a process.

Abstraction Transactions. There are four subclasses of abstraction transactions: judge, classify/decide, generalize, and transfer.

(1) A judge transaction requires a class frame with two or more subordinate instance frames. These frames can be entity, activity, or process frames. It enables the student to acquire the ability to order the instances of a given class on the basis of some dimension (criterion). The dimensions can be any attribute or combination of attributes.

(2) A classify/decide transaction requires a superclass frame with two or more subordinate class frames each of which have two or more instance frames. These frames can be entity, activity, or process frames. It enables the student to sort or classify instances as to class membership. It enables the student to select one alternative from another.

(3) A generalize transaction requires a superclass frame with two or more subordinate class frames each of which have two or more instance frames. These frames can be entity, activity, or process frames. Generalization transactions enable the student to combine instances of two or more classes into a more general class. Generalization is the inverse of classification.

(4) A transfer transaction requires a superclass frame and one or more class frames. These frames can be entity, activity or process frames. It enables the student to acquire an abstraction model, that is, a generalized set of steps for an activity, or a generalized set of events for a process, and to apply this abstraction model to a previously unencountered class or instance of the activity or process.

Association Transactions. There are five subclasses of association transactions: propagate, analogize, substitute, design, and discover.

(1) A propagate transaction requires two or more associated frames. The most common relations between knowledge frames -- uses, involves, applies -- all involve propagation. A propagation transaction facilitates the student's integration of information from two or more knowledge frames. One of the most important propagation associations is the link between an application activity and a tool activity; another is the link between a method activity and a process. Propagation enables the student to acquire one set of skills in the context of another set of skills.

While learning an application activity, the student can simultaneously learn the tool activity for doing the application. Or while learning a tool the student can simultaneously learn application activities for the tool. Or while learning a process the student can simultaneously learn a method activity for studying or observing the process. Or while learning a method activity the student can simultaneously learn the process for which the method was devised.

(2) An analogize transaction requires two or more knowledge frames linked by the relation analogous for. It enables the student to acquire the steps from one activity by likening it to an analogous activity; or to acquire the events in one process by likening it to an analogous process or activity.

(3) A substitute transaction requires two or more knowledge frames linked by the relation alternative for. It enables the student to learn an alternative activity or process by comparison, elaboration, or extension of a previously learned activity or process. It also enables the student to acquire alternative ways to accomplish a given activity or to explain a given process.

(4) A design transaction (Not required in XAIDA)

(5) A discover transaction (Not required in XAIDA)

The Transaction Manager for Maintenance Training. The transactions necessary to acquire all the knowledge and skill associated with a given enterprise comprise a transaction family. In maintenance, the six enterprises are (1) equipment operation, (2) equipment calibration and adjustment, (3) equipment testing, (4) access and disassembly, (5) equipment repair, and (6) troubleshooting.

The type and sequence of interactions necessary to acquire each of these complex enterprises is different; a high level transaction manager (TRXM) is required for each of the enterprises. The transaction manager is a program that calls and sequences the primary transactions identified as necessary for this curriculum.

In addition to transaction families for each of the six maintenance training enterprises, an equipment model family of transactions is a component of each of the other six transaction families.

(1) Transaction family for equipment operation enterprises. Two nested transaction families are required to operate a particular device or piece of equipment: Two classes of transactions are represented: execute - (6) equipment operation procedures; and decide/classify - (7) operation procedure or job aid selection.

(2) Transaction family for equipment calibration and adjustment enterprises. One nested transaction family is required to calibrate and adjust a particular device or piece of equipment. Three classes of transactions are represented: execute - (8) calibration and adjustment procedures; judge - (9) calibration and adjustment judgment; and decide/classify - (10) calibrate and adjust procedure or job aid selection.

(3) Transaction family for equipment testing enterprises. One nested transaction family is required to test a particular device or piece of equipment. Three classes of transactions are represented: execute - (11) testing procedures; judge - (12) testing judgment; and decide/classify - (13) test procedure or job aid selection.

(4) Transaction family for equipment access and disassembly enterprises. One nested transaction family is required to access and disassemble a particular device or piece of equipment. Two classes of transactions are represented: execute - (14) access and disassembly procedures; and decide/classify - (15) access and disassembly procedure or job aid selection.

(5) Transaction family for equipment repair enterprises. Two transaction shell instances and four nested transaction families are required to repair a particular device or piece of equipment. Two classes of transactions are represented: execute - (16) repair procedures; and decide/classify - (17) repair procedure or job aid selection.

(6) Transaction family for equipment trouble shooting enterprises. Four transaction shell instances and four nested transaction families are required to troubleshoot a particular device or piece of equipment. Two classes of transactions are represented: execute - (18) logical fault isolation procedures, and (19) intuitive fault isolation procedures; and decide/classify - (20) logical fault isolation procedure or job aid selection and (21) intuitive fault isolation procedure or job aid selection.

(7) Transaction family for equipment redesign and jury rig enterprises. Six transaction shell instances and three nested transaction families are required to design or jury rig a particular device or piece of equipment. Five classes of

transactions are represented: execute - (22) heuristic jury rig procedures; decide/classify - (23) redesign or jury rig procedure selection; transfer - and (24) procedural transfer; analogize - (25) conceptual and procedural analogies; substitute - (26) conceptual, functional, and procedural substitution; and design - (27) redesign techniques.

(8) Equipment Model TRX Family. Five transaction shells are required to construct a mental model of a particular equipment. Two classes of transactions are represented: identify - (1) physical & conceptual structure; and interpret - (2) device functioning, (3) device configuration, (4) fault recognition, and (5) prediction. The equipment model transaction family is not a "stand-alone" family, but is a component of the transaction family required for each of the other six maintenance training enterprises.

Customizing (Tailoring) Transaction Classes. Maintenance training utilizes nine of the twelve primary transactions: identify, interpret, execute, judge, decide/classify, transfer, analogize, substitute, and design. They are listed below with the transaction families in which they are employed. Four of nine maintenance transaction classes will be implemented in XAIDA: Identify, Interpret, Execute, and Decide/Classify.

Identify (XAIDA):

(1) Physical and conceptual structure

Interpret (XAIDA):

- (2) Device functioning
- (3) Device configuration
- (4) Fault recognition
- (5) Prediction

Execute (XAIDA):

- (6) Equipment operation procedures
- (8) Calibration and adjustment procedures
- (11) Testing procedures
- (14) Access and disassembly procedures
- (16) Repair procedures
- (18) Logical fault isolation procedures
- (19) Intuitive fault isolation procedures
- (22) Heuristic jury rig procedures

Judge:

- (9) Calibration and adjustment judging
- (12) Testing judging

Decide/Classify (XAIDA):

- (7) Equipment operation procedure or jobAid selection
- (10) Calibration and adjustment procedure or jobAid selection
- (13) Test procedure or jobAid selection
- (15) Access and disassembly procedure or jobAid selection
- (17) Repair procedure or job aid selection
- (20) Logical fault isolation procedure or jobAid selection
- (21) Intuitive fault isolation procedure or jobAid selection
- (23) Redesign or jury rig procedure selection

Transfer:

(24) Redesign or jury rig procedure transfer (Not required in XAIDA.)

Analogize:

(25) Redesign or jury rig analogies (Not required in XAIDA.)

Substitute:

(26) Redesign or jury rig substitution (Not required in XAIDA.)

Design:

(27) Redesign or jury rig redesign techniques. (Not required in XAIDA.)

A common base transaction shell will be constructed for each of the four TPX classes to be implemented in XAIDA. Each of the four TRX class shells will then be tailored to implement one or more of Halff's maintenance TRXs, as shown in Table 3.1. Interaction modes, interaction strategies, knowledge representation, and interaction parameters will vary depending on the family of which it is a part.

The following paragraphs describe the principal performance enabled by each of the nine primary transactions used in maintenance training, the knowledge base required for each, and the interaction modes required. The four transactions to be implemented in XAIDA are labeled.

Identify* (XAIDA):

(1) Physical and conceptual structure (XAIDA, YES)
Learning the names, location, and function of the parts of a device as a prerequisite to learning how the device works or how to operate it. Learn the meaning of a term and its position in a semantic network, e.g. CBESS and Tennyson's Milin.

Performance.

Students are shown images of the physical equipment and asked to identify individual components, their function, and their immediate connections. Students are shown diagrams representing the conceptual structure of the equipment and asked to identify individual components, their function, and their immediate connections. Students are shown both the physical and conceptual representations and asked to demonstrate the correspondence between these two representations.

Knowledge required.

The knowledge base includes some representation of the device, probably graphic, with the parts isolated and an associated name and function available.

A graphic representation of the conceptual representation of the system, pairing the conceptual symbols with the physical representation of the device.

Interactions.

The transaction must (1) present the names to the student and (2) enable the student to practice locating the parts and identifying the part name and function. The transaction must

present the conceptual names to the student, must pair the conceptual names with their physical referents. The transaction must enable the student to practice identifying conceptual symbols given referents, referents given the conceptual symbols, names of conceptual symbol given its graphic representation, and reproducing the symbol.

Interpret* (XAIDA):

(2) Device functioning (XAIDA, YES)

Performance.

Students are asked to discriminate among component states on the basis of some physical depiction of those states. The student can explain how the device functions, recognize the various states.

Performance.

Learner enters the sequence of events, and the conditions under which different events occur.

Knowledge required.

An operational, controllable simulation of the device with some degree of functional and structural fidelity.

Interactions.

The transaction must present the operation of the device via a simulation that enables the learner to change conditions and parameter values and observe the effects on the operation of the device. The transaction must be able to illustrate the operation of the device in a structured manner as well as enabling the learner to manipulate the operation.

(3) Device configuration (XAIDA, YES)

Performance.

Students are shown some of the inputs to an element of the device and asked how its other inputs must be set in order to achieve a desired function or state.

(4) Fault recognition (XAIDA, YES)

Performance.

Students are shown the actual outputs and inputs to an element and asked to determine whether or not the element is faulted.

(5) Prediction (XAIDA, YES)

Performance.

Students are given information about all inputs to a component or subsystem and required to predict the state of the component or subsystem, its outputs under normal operating conditions, and its outputs in each possible fault mode.

Execute* (XAIDA):

(6) Equipment operation procedures (XAIDA, YES)

Performance.

Students are required to perform certain operational functions using both a physical and conceptual simulator. That is, each step in the procedure must be executed within the physical

simulator and the conceptual simulator. For complex procedures the goal structure of the procedure should be tracked during procedure execution.

(8) Calibration and adjustment procedures (XAIDA, MAYBE)
Performance.

Students work with a physical simulation of the device to practice required calibration and adjustment tasks. A conceptual simulation of the system being adjusted or calibrated shows relations among the components involved in the process.

(11) Testing procedures (XAIDA, MAYBE)
Performance.

Students are required to carry out fixed testing procedures on a physical simulation of the equipment. A conceptual simulation of the components being tested is used to exhibit or query the student on the states of these components.

(14) Access and disassembly procedures (XAIDA, MAYBE)
Performance.

Students are given the task of gaining access to a particular component. They use a physical simulation of the device to practice the task. A matching conceptual simulation shows which components are accessible at each point in the procedure.

(16) Repair procedures (XAIDA, MAYBE)
Performance.

(18) Logical fault isolation procedures (XAIDA, YES)
Some forms of troubleshooting are best solved by acquiring an accurate operational model of the device or circuit that is malfunctioning and then systematically testing and or replacing components to eliminate potential trouble spots.

Performance.

Logical fault isolation enables the learner to acquire an accurate functional model of the device or circuit. The student acquires a set of systematic procedures for testing and/or replacing the components of the device or circuit. Students are provided with a conceptual simulation containing a single faulted component. At each point in the troubleshooting exercise, students choose an action and exhibit the consequences of the action.

Knowledge required.

An accurate logical operational model of the device or circuit. A set of malfunctioning components which can be inserted into the device or circuit. A functional simulation of the device or circuit.

Interactions.

The transaction must present the functional model of the device or circuit and enable the student to use this model in isolating faults that the system inserts into the functional model. The interactions must provide guidance which leads the student through systematic fault isolation activities.

(19) Intuitive fault isolation procedures (XAIDA, YES)
Logical fault isolation is often less efficient than the use of a set of heuristic guidelines (rules of thumb) to identify and correct faults in a device or circuit.

Performance.

Intuitive fault isolation enables the student to acquire a set of heuristic guidelines and to apply these guidelines in isolating a fault.

Knowledge required.

An accurate logical functional model of the device or circuit. A set of malfunctioning components which can be inserted into the device or circuit. A functional simulation of the device or circuit. A set of heuristic guidelines for troubleshooting the device or circuit.

Interactions.

The transaction must present the functional model of the device or circuit and enable the student to use this model in isolating faults that the system inserts into the functional model. The transaction must also present, and enable the student to acquire, the heuristic fault isolation rules. The interactions must provide guidance which enables the student to use the heuristics for fault isolation activities.

(22) Heuristic jury rig procedures

Performance.

Students are provided with conceptual simulations of tasks requiring complete or partial reconstruction of the equipment. For example, students could be required to restore as much functionality as possible with a limited inventory of spare parts or with other constraints on the reconstruction.

Judge:

- (9) Calibration and adjustment judging
- (12) Testing judging

Decide/Classify* (XAIDA):

- (7) Equipment operation procedure or jobAid selection (XAIDA, YES)
- (10) Calibration and adjustment procedure or jobAid selection (XAIDA, MAYBE)
- (13) Test procedure or jobAid selection (XAIDA, MAYBE)
- (15) Access and disassembly procedure or jobAid selection (XAIDA, MAYBE)
- (17) Repair procedure or job aid selection (XAIDA, MAYBE)
- (20) Logical fault isolation procedure or jobAid selection (XAIDA, YES)
- (21) Intuitive fault isolation procedure or jobAid selection (XAIDA, YES)
- (23) Redesign or jury rig procedure selection (XAIDA, NO)

Performance.

The following performance applies to transactions 7, 10, 13, 15, 17, 20, 21, and 23 listed above. Each of these transactions is a minor variation of the same decide/classify transaction. Students are asked to identify the procedures needed to deal with particular situations and to select any appropriate job aids.

Support is provided for this exercise in the form of subgoals and intermediate steps needed to arrive at the proper selection.

Transfer:

- (24) Redesign or jury rig procedure transfer (not required in XAIDA).

Analogize:

- (25) Redesign or jury rig analogies (not required in XAIDA).

Substitute:

- (26) Redesign or jury rig substitution (not required in XAIDA).

Design:

- (27) Redesign or jury rig redesign techniques (not required in XAIDA).

Shell Parameters. Each TRX shell has a number of parameters which configure the operation of the shell for a particular instructional instantiation. These parameters are set by the SME/ID. Parameters for a naming transaction for the components of an entity include:

(1) Focus Focus is a pointer to an entity frame in the EFN. The component hierarchy in which this frame participates will be instructed by the shell.

(2) Content Indicates how much of the component hierarchy is available for instruction. Takes one of the following (default is All):

All: entire hierarchy;

(list): a list of frames (subset of the hierarchy) which are the content.

(3) Coverage Indicates how much of the component hierarchy for the focus is to be instructed. Takes one of the following (default is Focus):

All: entire hierarchy;

Focus: the focus, its superpart (the single frame directly above the focus in the hierarchy), and its first level of subparts;

Levels: requires an additional integer argument, which indicates how many levels of subparts of the focus are instructed;

Exemplar, <label>: the focus and a single, specified subpart are instructed;

Random Exemplar: the focus and a single, randomly selected subpart are instructed.

(4) Guidance Level Sets the level of guidance to the student. Takes one of the following (default is Full):

None: no guidance;

OnDemand: guidance presented on student request only;

Full: guidance at all times;

Faded: begin with full guidance, fade to OnDemand by end of transaction.

(5) Guidance Type Takes one of two values (default is Verbose):

Concise: guidance interactions are short and to the point;

Verbose: guidance interactions are detailed and complete.

(6) View Representation of the subject matter. Takes one or more of the following (default is Structural):

Structural: displays the component relation in terms of the knowledge structures, in tree format;

Physical: displays an author-supplied graphic or illustration of the object whose parts are being instructed, representing the physical appearance of the object;

Functional: displays an author-supplied graphic or illustration of the object whose parts are being instructed, representing the functional appearance of the object.

(7) Vertical Sequence Order of introduction of the parts. Takes one of the following values (default is TopDownBreadth):

TopDownBreadth: ordering is from the highest superpart to the lowest level, breadth first (an entire level is introduced before any components of the next level are introduced);

TopDownDepth: ordering is from the highest superpart to the lowest level, depth first;

BottomUp: ordering is from the lowest subpart level to the highest, breadth first.

(8) Temporal Sequence Within the vertical sequence, is the order of introduction of the parts on the same level. The default is LeftRight:

LeftRight: ordering is according to the representation in the EFN, from left to right;

LowHigh, <attribute label>: ordering is according to the ranking of a named attribute, from low to high;

HighLow, <attribute label>: ordering is according to the ranking of a named attribute, from high to low.

(9) Trials The number of times to sequence through the set of parts. Takes a positive integer value, default is 1.

(10) Mastery Level The percent correct for mastery. Takes a value between 0 and 100 (default is 80%).

(11) Response Mode Type of response required of the student. Takes one of two values: Recognition or Recall (default is Recognition).

(12) Feedback Timing of feedback. Takes one of the following values (default is PrePractice):
None: no feedback is given;
PostResponse: corrective feedback is given immediately after a response;
PrePractice: corrective feedback is given just before the next opportunity to practice.

(13) Replacement Whenever sampling is used in the transaction, this parameter controls whether a new sample may or may not include items from previous samples (default is With).
With: a new sample may include items previously used;
Without: new samples are distinct from previous samples.

(14) Items Whenever a pool of items is practiced or tested, this parameter sets the maximum size of the pool. It takes a positive integer value (default is 3).

(15) Timeout The amount of time to wait for a student response before timing out. If the student response involves typing rather than pointing, the timeout occurs after a base interval, plus a fraction of the base interval multiplied by a number derived from the length of the expected response. If the student response is pointing only, the timeout occurs after the base interval. The value of the parameter is the base interval, a positive integer (default is 3 (seconds)).

(16) Item Order Whenever a pool of items is practiced or tested more than once, this parameter controls whether the ordering is the same or different (default is Random).
Random: the ordering is random;
Same: the ordering is fixed.

(17) Mode The mode is the method of interaction with the student. One or more of the following may be selected (default is Overview):

Overview: presents the knowledge structure from the knowledge base in the Structural view;
Presentation: presents an author-supplied graphic in either the Physical or the Functional view, and demonstrates the parts to the learner;
Practice: provides practice for the learner using the author-supplied graphic, in either the Physical or Functional view;
Instance
Assessment: tests the student's mastery of the material, using the author-supplied graphic, in either the Physical or Functional view.

(18) Strategy Strategy is defined as a sequence of modes. Because modes are fully determined by strategy, the arguments to the Mode parameter are ignored unless the Strategy parameter is None (the default).
Overview: the Overview mode;

Familiarity: the Overview mode followed by the Presentation mode;
 Basic: Overview plus Presentation plus Practice modes;
 Mastery: Overview plus Presentation plus Practice plus Instance Assessment;
 BasicRemediation: Instance Assessment, followed by Basic Strategy to remediate errors;
 MasteryRemediation: Instance Assessment, followed by Mastery Strategy to remediate errors;
 Assessment: Instance Assessment mode;
 Summary: Overview with Coverage set to Focus, plus Presentation followed by Instance Assessment, with Coverage set to Random Exemplar for both;
 None: no strategy.

(19) Strategic Control Determines the level of control granted the student over the selection of strategy, mode, and content. Takes one of the following (default is System):
 System: the strategy, mode, content, and coverage are delivered as set by the parameter values;
 Student: the student may select alternate strategies, mode, content, and coverage.
 (Dynamic instructor override.)

(20) Tactical Control Determines control over the initiation and termination of interactions. Also determines whether the student may alter the values of the Guidance, View, and Sequence parameters. Takes one of two values, System or Learner (default is System).

Configuring Shells. Configuring is the setting of parameters to a shell, and attaching content.

For example, a call to a shell in the Identify class to instruct the names and locations of the left engine starting circuitry might be configured as follows:

focus: left engine starting circuitry
 content: all
 coverage: all
 guidance level: faded
 guidance type: concise
 view: structural, functional
 vertical sequence: topDownBreadth
 temporal sequence: leftRight
 trials: 1
 mastery level: 100%
 response mode: recall
 feedback: postResponse
 replacement: without
 items: 3
 timeout: 3
 item order: random
 strategy: basic
 strategic control: system
 tactical control: learner.

Detailing Shells. Detailing is the attachment of graphics and text, prepared offline, to the knowledge base for use by the shells. With the use of simulations, detailing requirements are replaced in large measure by simulation authoring.

3.2.1.4 Courseware Generation

Delivery of instruction to a large number of students may not be a function of the AIDA system. Instruction delivery is more likely to be accomplished via a multi-terminal, multi-media system optimized for instruction delivery, such as ISS, WISE, QUEST, PILOT, TENCORE, etc. The Courseware Generation System (CGS) analyses the configured and authored transactions and generates courseware in a format that can be used by other instruction delivery systems. CGS will not be required in XAIDA. XAIDA will, however, have a student mode which will enable the SME/ID to evaluate an instruction sequence.

3.2.1.5 Instruction Delivery

The fourth mode of operation of AIDA is instruction delivery as though to a student. This section describes the delivery of instruction to the student, the output of the system. Delivery of instruction to a large number of students may not be a function of the AIDA system. Instruction delivery is more likely to be accomplished via a multi-terminal system optimized for instruction delivery, such as ISS, WISE, QUEST, PILOT, TENCORE, etc. AIDA will, however, have a student mode which will enable the SME/ID to evaluate an instruction sequence.

Instructional Modes and Strategies. The type of transaction and the components of its knowledge base limit the interactions that are possible within a given transaction shell. Different classes of transactions will have different types of interactions. Nevertheless, all transactions should include interactions that are characterized by certain interaction modes. Interaction mode alternatives determine the method of interaction with the student. Interaction modes assume different values on the form of the interaction (expository or inquisitory) and the degree of learner control involved (learner control or system control). Five interaction modes have been identified: (1) overview, (2) presentation, (3) practice, (4) generality assessment, and (5) instance assessment.

(1) Overview mode presents the knowledge structure, as represented in the EFN. For example, in an Identify transaction, overview shows the parts hierarchy of an entity in tree format. Text instruction may accompany the diagrams. The overview serves as an advance organizer, and as a review.

Example Overview mode for an entity component hierarchy.

(2) Presentation mode demonstrates and presents the content represented by the knowledge structure, in terms of both generalities and instances. For example, an Interpret transaction for device operation simulates the operation of the device, explaining the events associated with the process.

Example Presentation mode for an Identify transaction.

Example Presentation mode for an Interpret transaction.

(3) Practice mode provides opportunity for the learner to work with the content directly. For example, an Execute transaction for an activity provides a simulation which can be manipulated by the student, with the consequences of actions displayed. Practice for an Interpret transaction allows the student to adjust controls, regulate inputs, and modify the functioning of devices, and to predict the consequences of these actions.

Practice mode for an Interpret Shell.

(4,5) Generality and instance assessment modes test at the generality and instance level, respectively. Test results are recorded by the delivery system, under parametric control of the transaction shell.

Interaction strategy is the combination and sequence of interaction modes available to the student. There are eight interaction strategies: (1) overview, (2) familiarity, (3) basic, (4) mastery, (5) basic remediation, (6) mastery remediation, (7) summary, and (8) assessment.

(1) Overview consists of the overview interaction mode.

(2) Familiarity consists of an overview interaction plus a presentation.

(3) Basic instruction consists of an overview plus presentation plus practice.

(4) Mastery instruction consists of overview plus presentation plus practice plus generality and/or instance assessment; if the criterion is not met, a new presentation, practice, and assessment for missed items is engaged until the criterion is met.

(5) Basic remediation consists of generality or instance assessment; if the criterion is not met, then basic instruction is provided for the missed items.

(6) Mastery remediation consists of generality or instance assessment; if the criterion is not met, then mastery instruction is provided for the missed items until the criterion is met.

(7) Summary is an overview plus presentation followed by instance assessment; both the presentation and the assessment are for a single representative element of the knowledge structure, rather than the full knowledge structure.

(8) Assessment consists of generality or instance assessment.

Interaction of Simulations and Shells. In order to integrate the shells with the simulations at delivery time, there must be a communications interface established. This interface establishes conventions whereby shells are able to:

- o query simulations to determine their capabilities
- o query status information from simulations
- o issue commands to simulations to set the simulation directly into a state
- o replace direct learner input with commands from the shell.

On-line Delivery Advisor. The authoring decisions made at design time are based on the designer's best estimate of the student population. During the delivery of instruction, information about the student -- aptitude, specific goals, motivation, familiarity, and other factors, as well as the student's expressed preferences -- may be taken into account to modify those decisions. An on-line delivery advisor has access to the domain knowledge base and the configurations. In addition, it maintains a student model that contains information about the student. Using the information gathered about the student, the advisor adjusts design decisions to customize the instruction to more adequately meet the characteristics of the student. The advisor also can engage in a mixed-initiative dialogue with the student which allows the student to participate in this decision-making.

3.2.1.6 Evaluation

Dr. O'Neil declared that AIDA needs both author management and course management systems. Without computer-managed instruction there can be no benchmarks with which to determine progress in the evolution of the AIDA system, and there will be weak USAF support.

Furthermore, if XAIDA is to be a research tool, then measurement and evaluation issues must be confronted directly by adding a data gathering capability. The built-in evaluation system will enable users to turn in data to be used in refining AIDA. Data will be collected on both instructors and students.

3.2.2 **System Capability Relationships**

3.2.2.1 Major Components for AIDA

The USAF TTCs currently use Zenith Z-248 80286 microcomputers with EGA monitors for CBI, running under Microsoft DOS 3.0. As a result of the Desktop III acquisition, it would seem likely that these will be gradually upgraded to 80386 systems with higher resolution (VGA/SuperVGA) monitors and additional multi-media peripheral devices (CD-ROM, interactive videodisk, etc.), running

1. Microsoft Windows 386 under Microsoft DOS,
2. X Windows under UNIX.
3. IBM Presentation Manager under OS/2, and /or

AIDA will include the following major components, subject to subsequent changes due to an evolutionary design:

(1) A COTS hypermedia product which satisfies the requirements for multi-media CBI design, development and delivery, presents the same consistent user interface of pull-down menu, icon, and button driven processing of the same input files across the proposed industry-standard graphical user interfaces (GUIs) on the proposed platform, and provides an "open architecture", with "black box" software slot modules for additional peripherals currently available and to be developed in this rapidly changing field.

(2) An 80386 hardware platform with:

1. High resolution color graphics input/output, including Monitor(s), Scanner, Printer, and Projector
2. High capacity rapid access erasable optical disk
3. Interactive videodisk
4. CD-ROM
5. High fidelity audio input/output, including
6. Speech synthesis and recognition
7. Such additional standard input devices as the Mouse, Stylus, Joystick, and Touch-sensitive screen
8. New technologies at the R&D stage to be announced during the course of this three-year effort.

(3) Additional peripherals to support system functionality as required.

3.2.2.2 Common Capability

3.2.3 **External Interface Requirements**

3.2.4 **Physical Characteristics**

An AIDA development work station will support at least two simultaneous users.

3.2.4.1 Protective Coatings (N/A)

This section is not applicable to this specification.

3.2.5 **System Quality Factors**

3.2.5.1 Reliability

AIDA shall be designed with high quality equipment to provide a high level of reliability. AIDA shall:

- a. Be capable of performing full system operations except when power interruption occurs.
- b. Ensure that database file and record integrity is maintained for multiple users.

3.2.5.2 Maintainability

AIDA shall:

- a. Be modular in design with easily accessible line replaceable units (LRUs) and assemblies to minimize maintenance and system downtime.
- b. Keep the use of special tools and equipment for testing and maintenance to a minimum.

3.2.5.3 Availability (N/A)

This section is not applicable to this specification.

3.2.5.4 Additional Quality Factors (N/A)

This section is not applicable to this specification.

3.2.6 **Environmental Conditions**

AIDA shall withstand the following environmental conditions during operation:

- a. Normal high and low temperature of an office environment.
- b. High and low humidity of an office environment.
- c. Standard commercial power supplied at the site including the following:
 - (1) Voltage: 90 V - 130 V or 180 V - 260 V.
 - (2) Frequency: 47 Hz - 63 Hz.
- d. Meet power line transient requirements of ANSI/IEEE C62.41-1980 as a minimum.

3.2.7 **Transportability**

All bulky components shall be configured to enable easy transportability.

3.2.8 **Flexibility and Expansion**

AIDA shall take into consideration the following areas of growth to achieve system flexibility and expansion:

- a. Permit additional standardized equipment to be added by inserting appropriate interfaces and modifying only system parameters in the software.
- b. Accommodate larger databases and additional users.
- c. Be capable of providing more than one authoring station with networking capability.
- d. Stress the capability for tailoring the configuration of the system for unique installations.
- e. Utilize common commercial and government interface standards and common, readily available interface hardware to the extent possible.

3.2.9 Portability

Each component of the system shall be portable with normal handling.

3.3 Design and Construction

3.3.1 Materials (N/A)

3.3.2 Electromagnetic Radiation (N/A)

3.3.3 Nameplates and Product Marking (N/A)

3.3.4 Workmanship (N/A)

3.3.5 Interchangeability (N/A)

3.3.6 Safety (N/A)

3.3.7 Human Engineering

The Contractor shall take into consideration the guidance in MIL-STD-1472C to maximize the effectiveness of man/machine interfaces and to minimize operator reaction times, actions, and training requirements.

3.3.8 Nuclear Control (N/A)

This section is not applicable to this specification.

3.3.9 System Security

All hardware components making up the AIDA and the documentation provided are unclassified. The system will provide log on and password protection.

3.3.10 Government Furnished Property Usage

AIDA shall incorporate Government Furnished Property into the system, to be determined by the contract monitor as necessary.

3.3.11 Computer Resource Reserve Capacity

Computer resource reserve capacity shall be in accordance with the following:

- a. All memory, timing, bus loading and input/output must be adequate to provide spare and growth capability of no less than 100% above the spare memory installed for preplanned improvements and changes.
- b. Spare capacity must be spread evenly throughout the system, but not simply a system average.
- c. At installation, 67 percent of delivered memory capacity and 50 percent of CPU timing must be spare (that portion of the delivered memory and CPU timing unused when processing the worst case design load under single failure conditions).
- d. Spare memory must be allocated in consecutive memory blocks.
- e. All system busses and timing must be able to accept this increase in processing capacity without modification.
- f. Central processor must also have a 100 percent growth capability in delivered memory and throughput at the time of installation.

3.4 Documentation

The Contractor shall provide the following documentation in accordance with DOD-STD-2167A and DOD-STD-2168 if applicable:

- a. Commercial Documentation:
 - (1) Commercial technical manuals for all hardware used.
 - (2) Commercial technical manuals for software operation.
- b. New Documentation:
 - (1) Program Plan.
 - (2) AIDA System/Segment Specification.
 - (3) System Architecture/Preliminary Hardware Specification.
 - (4) Software Modification Plan/Preliminary Software Specification.
 - (5) AIDA Training Lesson Plan.
 - (6) Test Plan.
 - (7) Applicable engineering drawings.

3.5 Logistics

The USAF will plan for organic maintenance at the organizational level. Organizational-level maintenance will be conducted in the operational environment and will consist of removal and replacement of LRUs by appropriately trained personnel.

The contractor shall be responsible for the following logistics considerations:

- a. Software support for the duration of the contract.
- b. Hardware support for contractor supplied items.
- c. Maintenance training for organizational level personnel at the FTD.

3.6 Personnel and Training

3.6.1 Personnel

Personnel necessary for implementing AIDA is integral to a completely functioning system. Recommended personnel are (1) a Hardware System Specialist, and (2) a Software System Specialist. The minimum responsibility of each position is described as follows:

1. Hardware System Specialist - Installing and maintaining hardware components.
2. Software System Specialist - Installing, maintaining, and training software components.

3.6.2 Training

The contractor shall develop lesson plans for the initial on-site training of AIDA maintenance and operator personnel whose grade is E4 to E6. Operator personnel will have a background as an SME in the targeted technical field. The training will include up to 2 weeks of classroom lessons and on-the-job training (OJT) per year. Follow-on training will generally be conducted at the unit level through OJT by the trained software system specialist.

Operator personnel shall receive detailed instruction in operating procedures for use of the system.

3.7 Characteristics of Subordinate Elements

3.8 Precedence

The precedence of requirements within this specification shall be in the following order of priority:

- a. Performance Characteristics (3.2.1).
- b. System Capability Relationships (3.2.2).

- c. System Quality Factors (3.2.5).
- d. All other paragraphs of this specification.

3.9 Qualification

Each system requirement shall be approved for completeness with the following verification methods:

a. Inspection (I): This verification method is based on a visual examination to prove that specified requirements have been met.

b. Analysis (A): This verification method is based on proving, by induction or deduction, that specified requirements have been met. It relies upon a technical analysis and evaluation of data, equations, graphs, circuit and timing diagrams, coding or computer simulations. This method is generally applicable where the design uses hardware or software which is available off-the-shelf and for which data are readily available. In addition, analysis is applicable for new design where a more comprehensive method shall be used for verification in a subsequent program phase. Analysis may also be used in lieu of alternate methods of verification where significant cost savings would result.

c. Demonstration (D): This verification method does not normally use any special instrumentation or simulated inputs to prove that specified requirements have been met. This method shall be based on one of the following:

- (1) An operation, movement, or adjustment which relies on observation of an action or series of actions and the consequent results (e.g., on a CRT display or printout) to decide on the success of the demonstration.
- (2) A statistical review of a number of demonstration examples performed under specified conditions to determine compliance at a specified confidence level (e.g., reliability and maintainability demonstrations).

d. Test (T): This verification method normally involves the use of special instrumentation, simulated inputs or a combination of both to prove that specified requirements have been met. This method is generally applicable where specified external stimuli (hardware or software) and/or external devices are required to produce and/or measure (record) predictable result.

3.10 Standard Sample (N/A)

This section is not applicable to this specification.

3.11 Preproduction Sample, Periodic Production Sample, Pilot or Pilot Lot (N/A)

This section is not applicable to this specification.

4. QUALITY ASSURANCE PROVISIONS

Each system requirement in Sections 3 and 5 shall be tested for quality assurance with the verification methods specified in Section 3.9.

4.1 Responsibility For Inspection

4.1.1 Contractor's Responsibility for Inspection

The Contractor shall be responsible for performance of inspections at the AIDA site at ALHRD as follows:

- a. Assemble and interconnect the AIDA hardware received according to the site plans.
- b. Operationally test each AIDA component and rectify all malfunctions.
- c. Install the operating system.
- d. Install the AIDA software and rectify all malfunctions.
- e. Connect and individually test each unit that is not furnished with the AIDA for compatibility according to the external interface requirements, and rectify all malfunctions.
- f. Conduct a total integration test in accordance with the Test Acceptance Procedure (TAP).
- g. Notify the USAF that the AIDA is ready for testing.

4.1.2 Government's Responsibility for Inspection

The USAF shall be responsible for performance of inspections at the AIDA site at ALHRD as follows:

- a. A USAF representative must witness the entire process of the demonstration according to the TAP.
- b. This USAF representative will sign off the TAP if the demonstration is successfully completed in accordance with the TAP.

4.2 Special Tests and Examinations

The Contractor shall report the results of a two-week formative test which is conducted at one site at the end of each year.

4.3 Requirements Cross-Reference

This section is not applicable to this specification.

5. PREPARATION FOR DELIVERY

AIDA will be deployed inside Continental United States. The equipment required for installation of AIDA shall be packed to provide protection against the conditions characteristic of domestic shipment, handling, and storage.

6. NOTES

The following is a list of acronyms and abbreviations used throughout this specification.

AIDA	Advanced Instructional Design Advisor
ALHRD/IDC	Armstrong Lab (Human Resources Directorate)
AFR	Air Force Regulation
COTS	Commercial Off-the-Shelf
CPU	Central Processing Unit
CRT	Cathode Ray Tube
CSCI	Computer Software Configuration Item
DOD-STD	Department of Defense Standard
FCC	Federal Communications Commission
GFE	Government Furnished Equipment
IOC	Initial Operational Capability
ISD	Instructional System Design
LRU	Line Replaceable Unit
MIL-STD	Military Standard
NDI	Non-Developmental Item
OJT	On-the-job Training
TAP	Test Acceptance Procedure
TRX	Transaction
TRXS	Transaction Shell
UL	Underwriters Laboratories
USAF	U.S. Air Force
XAIDA	Experimental Advanced Instructional Design Advisor

6.1 Intended Use

6.1.1 Missions

The purpose of the AIDA is to provide intelligent and automated assistance to subject matter experts (SME) and instructional designers (ID) throughout all phases of instructional system development (ISD).

6.1.2 Threat

The USAF conducts a vast amount of training in technical areas that undergo frequent revisions and increases in complexity. USAF instructional designers must be able to rapidly incorporate revisions and complexity into instructional materials without increases in production time or decreases in student performance. The USAF has begun to use computers and other interactive technologies to meet the demands of rapid change and increased complexity in technical domains. Few guidelines for the effective and efficient use of these technologies are available, however, because most instructional design principles were developed prior to their existence. The research focuses on the formulation and testing of instructional design principles. It will identify and validate instructional design strategies, principles and prescriptions that exploit the capabilities of

interactive technologies (e.g., DVI, IVD, CD-ROM, CD-ROM/XA, CD-I, etc.) to enhance learning and retention of complex training materials. Of particular interest are methods of effectively and efficiently guiding instructional developers, especially novices, through the process of designing and developing instructional materials.

APPENDIX D
DATA BASE DESIGN DOCUMENT

Table of Contents

1 SCOPE	88
1.1 Identification	88
1.2 Purpose	88
1.3 Introduction	88
2 REFERENCED DOCUMENTS	89
2.1 Government Documents	89
2.1.1 Specifications	89
2.1.2 Standards	89
2.1.3 Other Publications	89
2.2 Non-Government Documents	90
3 REQUIREMENTS	91
3.1 Knowledge Base Management System (KBMS) Overview	91
3.1.1 Knowledge Base Manager (KBM)	91
3.1.2 Knowledge Base Definition Language (KBDL)	91
3.1.3 Knowledge Base Query Language (KBQL)	92
3.2 Knowledge Base Structure	93
3.2.1 Knowledge Base Structure Description	93
3.2.2 Knowledge Base Interrelationships	96
3.2.2.1 Domain Knowledge Base (DOMAIN.KB)	96
3.2.2.2 Transaction Knowledge Base (TRANSACTION.KB)	101
3.2.2.3 Enterprise Knowledge Base (ENTERPRISE.KB)	102
3.2.2.4 Student Knowledge Base (STUDENT.KB)	103
3.2.2.5 Environment Knowledge Base (ENV.KB)	104
3.2.2.6 Task Knowledge Base (TASK.KB)	104
3.2.2.7 Session Knowledge Base (SESSION.KB)	107
3.3 Knowledge Base Design	107
3.3.1 KB: Domain Knowledge Base	107
3.3.1.1 FRAME: Domain Knowledge Frame	107
3.3.1.2 FRAME: Entity Attributes Frame	108
3.3.1.3 FRAME: Activity Attributes Frame	109
3.3.1.4 FRAME: Process Attributes Frame	109
3.3.1.5 FRAME: Components Frame	110
3.3.1.6 FRAME: Abstraction Frame	110
3.3.1.7 FRAME: Association Frame	111
3.3.2 KB: Transaction Knowledge Base	111
3.3.2.1 FRAME: Transaction Frame	111
3.3.3 KB: Enterprise Knowledge Base	118
3.3.3.1 FRAME: Enterprise Frame	118
3.3.4 KB: Student Knowledge Base	122
3.3.4.1 FRAME: Student Profile Frame	122
3.3.5 KB: Environment Knowledge Base	123
3.3.5.1 FRAME: Environment Profile Frame	123
3.3.6 KB: Task Knowledge Base	124
3.3.6.1 FRAME: Task Frame	124
3.4 Knowledge Base and Library References	124

Table of Contents (Cont'd)

4 UNSPECIFIED IN DI-MCCR-80028	124
5 UNSPECIFIED IN DI-MCCR-80028	124
6 NOTES	125
6.1 Background Information	125
6.2 Glossary	126
6.3 Acronyms and Abbreviations	128

Figures

Figure D-1: AIDA System/Knowledge Base overview	93
Figure D-2: Component hierarchy diagram	98
Figure D-3: Elaborated Frame Network (EFN) example diagram	100

1. SCOPE

1.1 Identification

This Data Base Design Document (DBDD) describes the detailed design of the data base identified as the Knowledge Base (KB) for the Computer Software Configuration Item (CSCI) identified as the Advanced Instructional Design Advisor (AIDA) expert system. The AIDA expert system itself is under specification by Mei Associates, Inc., for the Armstrong Laboratory, Human Resources Directorate (ALHRD/IDC) in accordance with DI-CMAN-80008A, System/Segment Specification (SSS).

1.2 Purpose

The AIDA.KB serves as the repository of the knowledge required to support the AIDA expert system (ES) in enabling technical subject matter experts (SMEs) to produce effective computer-assisted instruction in their specialty areas, even though they may have little or no expertise in the field of instructional design.

As currently conceived, the AIDA Knowledge Base (AIDA.KB) consists of the following sub-KBs:

- a. Domain Knowledge Base (DOMAIN.KB)
- b. Transaction Knowledge Base (TRANSACTION.KB)
- c. Enterprise Knowledge Base (ENTERPRISE.KB)
- d. Student Knowledge Base (STUDENT.KB)
- e. Environment Knowledge Base (ENV.KB)
- f. Task Knowledge Base (TASK.KB)
- g. Session Knowledge Base (SESSION.KB)

1.3 Introduction

This DBDD serves to communicate and control data base design decisions to the Government. Upon completion of Physical Configuration Audit (PCA), this DBDD (and/or its successor documents) will become a part of the Product Baseline for the AIDA CSCI.

Section 3 describes the design of the AIDA Data or Knowledge Base. Section 3.1 is an overview of the Knowledge Base Management System (KBMS), including the component Knowledge Base Manager (KBM), the Knowledge Base Definition Language (KBDL), and the Knowledge Base Query Language (KBQL).

Section 3.2 contains a description of the Knowledge Base Structure and the Knowledge Base Interrelationships, including the Domain KB, the Transaction KB, the Enterprise KB, the Student KB, the Instructional Environment KB, the Task KB, and the Session KB.

Section 3.3 explains the design of each of the knowledge bases described in Section 3.2, defining each frame, slot, and facet.

2.0 REFERENCED DOCUMENTS

2.1 Government Documents

2.1.1 Specifications

DI-CMAN-80008A, System/Segment Specification, 29 February 1988.

2.1.2 Standards

- a. DOD-STD-2167A, February 29, 1988, Defense Systems Software Development.
- b. DOD-STD-2168, Defense Systems Software Quality Program.
- c. MIL-STD-499A, Engineering Management.
- d. DOD-HDBK-287, Defense System Software Development.
- e. MIL-STD-490A. Military Standard. Specification practices, AMSC No. F3630. 4 June 1985 [Superseding MIL-STD-490 dated 30 October 1968.]
- f. MIL-STD-1521B. Technical Reviews and Audits for Systems, Equipments, and Computer Software. AMSC 3631. 4 June 1985. [Superseding MIL-STD-1521A (USAF) dated 1 June 1976.]
- g. MIL-STD-1472C. Human Engineering Design, Criteria for Military Systems Equipment and Facilities. Notice 2, 12 May 1981.

2.1.3 Other Publications

Data Item Description (DID) DI-MCCR-80028, Database Design Document

RAPIDS/TRX Walkthrough, AIDA Technical Interchange Meeting Presentation by M. David Merrill and Mark K. Jones, October 1990.

Research in Instructional Design for Interactive Technologies, Volume I - Technical Proposal, July 16, 1990.

Specifications for an Advanced Instructional Design Advisor (AIDA) for Computer-Based Training, Final Report, Contract No. F33615-88-C-0003, Task Order No. 0006, 31 July 1990.

System Specification for the Experimental Advanced Instructional Design Advisor (XAIDA), CDRL Seq. No. 31, Contract No. F33615-88-C-0003, Task Order No. 0006, 15 May 1991.

Merrill, M., Li, Z., and Jones, M., Project AIDA: A Concept Paper, Sep 1989.

Merrill, M.D., Project AIDA: A Theory Paper, Sep 1989.

Halff, H.M., Automating Maintenance Training, Jul 1990.

Halff, H.M., Teaching Procedures, Nov 1990.

Merrill, M.D., ITX for Maintenance Training, Sep 1990.

Merrill, M.D. and Jones, M.K., RAPIDS/TRX Walk-through, Oct 1990.

Halff, H.M., Automating Instructional Design and Development, Feb 1990.

Hickey, A.E., Proceedings of the Working Group Meeting at USU 4-6 Sep 1990, 18 Sep 1990.

Hickey, A.E., Proceedings of the Working Group Meeting at San Antonio, TX, 14-16 Jan 1991.

2.2 Non-Government Documents

ANSI/IEEE Std. 729-1983, IEEE Standard Glossary of Software Engineering Terminology.

IBM, Systems Application Architecture, Common User Access Advanced Interface Design Guide, First Edition, June 1989.

Giarratano, Joseph; Riley, Gary: Expert Systems: Principles and Programming. Boston, MA: PWS-KENT Publishing Company, 1989.

Marshall, Garry: Advanced Students' Guide to Expert Systems. Oxford, England: Heineman Newnes, 1990.

3. REQUIREMENTS

3.1 Knowledge Base Management System (KBMS) Overview

A knowledge base management system (KBMS) consists of software for storing, accessing, manipulating, reasoning, and otherwise controlling the knowledge embodied in an expert system. In the case of AIDA, the KBMS will manage (1) the knowledge that defines and drives the instruction configuration and authoring functions, (2) the instruction configuration and authoring knowledge acquired from subject matter experts (SME) and instructional designers (ID), and (3) the knowledge that defines and drives the instruction delivery functions.

Instructional design knowledge, both accumulated and to be elicited, is normally stored in prose forms suitable for humans to consult, understand, and apply to the process of course development. In the AIDA.KB, however, this knowledge must be appropriately structured for processing by a computer utilizing expert system technology. As summarized in Marshall (1983), an acceptable representation for structured knowledge should have the following characteristics:

- a. representational adequacy: It should support the acquisition of all the aspects of the knowledge in all their subtlety.
- b. representational efficiency: It should allow efficient acquisition so that the knowledge is stored compactly and is easily accessible.
- c. inferential adequacy: It should be possible to use the knowledge in any way that may be appropriate.
- d. inferential efficiency: The knowledge should be located and used rapidly and without the need of excessive computation.

3.1.1 Knowledge Base Manager (KBM)

The Knowledge Base Manager (KBM) is the core component of the KBMS. It controls the base level storage of the knowledge base.

3.1.2 Knowledge Base Definition Language (KBDL)

The Knowledge Base Definition Language (KBDL) is the "front-end" syntax used to define/declare the objects in the knowledge base. Of the various approaches to representing structured knowledge, the one proposed for the AIDA KBMS is a generalized, object-oriented (OOD) approach using frames, not scripts, as the objects.

The frame was developed from the observation that people do not construct their ideas about familiar objects, situations, and events from scratch, but carry with them a (potentially very large) set of expectations about these things. Frames have proven useful in representing what is often referred to as "common

sense" knowledge, dependent as such knowledge is on many such unarticulated ("default") expectations.

An object-oriented approach, using frames as objects, has been shown to be appropriate for representing mechanical devices, such as automobiles, and therefore suitable for the designated AIDA task area of equipment maintenance. As may be surmised from the discussion of an object oriented KB structure above, this approach can satisfy the requirements for representational and inferential adequacy.

AIDA will employ a KBDL developed by Mei Associates and approved by ALHRD. Upon approval of the KBDL, this paragraph and the descriptions of frames in subsequent paragraphs will be revised to specify the KBDL. However, for convenience of reference, the following general terms are herewith defined for use in the description of the AIDA knowledge base structure:

- a. knowledge base (KB): The structured knowledge stored in an expert system. In AIDA, a named collection of related frames viewed as a unit.
- b. object: A data structure that contains all the information related to a particular entity. It might be considered a frame with additional features allowing it to contain and invoke methods and to send and receive messages.
- c. frame: A named set of one or more related slots; equivalent to a record in conventional data base terminology. Frame is the object in this OOD.
- d. slot: A named set of one or more related facets; sometimes equivalent to a data field in conventional data base terminology.
- e. facet: A variable or a parameter name designating an attribute (value, constraint, link, procedure, etc.) of a slot; sometimes equivalent to a data item in conventional data base terminology.
- f. value: The lowest level item of knowledge stored in a KB.

3.1.3 Knowledge Base Query Language (KBQL)

In a KBMS, the Knowledge Base Query Language (KBQL) is the "front-end" syntax used to insert, retrieve, update, and delete the objects in a knowledge base. AIDA will employ a KBQL developed by Mei Associates and approved by ALHRD. Upon approval of the KBQL, this paragraph will be revised to specify the KBQL.

3.2 Knowledge Base Structure

3.2.1 Knowledge Base Structure Description

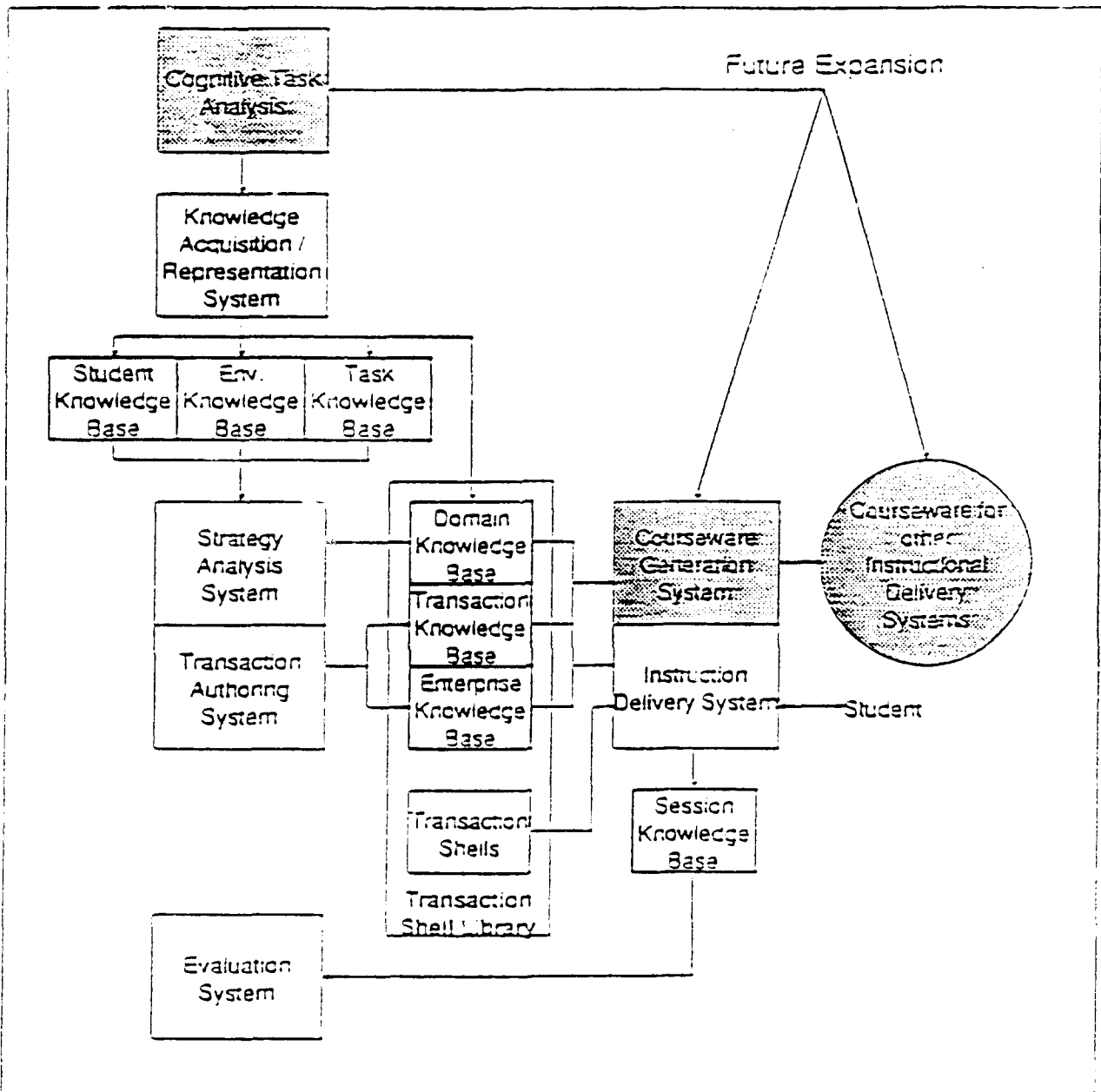


Figure D-1: AIDA System/Knowledge Base overview.

AIDA is comprised of 6 major subsystems and 7 knowledge bases as illustrated in Figure D-1. The Knowledge Acquisition/Representation, Strategy Analysis, Transaction Authoring and Instruction Delivery systems closely correspond to the steps in the course development process: i.e., the AIDA concept/process as defined in AIDA Phase I. The Evaluation System further supports the cyclic nature of this process by providing the capability to analyze an instance of a configured and authored

course or lesson. And finally, the Courseware Generation System is available to generate courseware for instruction delivery systems external to AIDA at some future date (e.g., ATS, ISS, MERLIN, etc.).

- a. The Knowledge Acquisition/Representation System (KARS) interacts with the subject matter expert (SME) and/or the instructional designer (ID) to gather information on (1) the task to be learned, (2) the student who must learn this task, (3) the environment in which the student will be instructed, and (4) a model of the subject matter associated with the task to be learned, as well as supporting instructional material such as figures, diagrams, and descriptions. This knowledge is stored in the Task, Student, Environment, and Domain Knowledge Bases. The KARS may provide the capacity to accept inputs from an external cognitive task analysis system at some future date, e.g., the ALHRD/MO cognitive task analysis system.
- b. The Strategy Analysis System analyzes the content of Student, Environment, Task, and Domain Knowledge Bases, then invokes the Transaction Authoring System to interact with the SME/ID to specify the approach, organization, and additional content of the curriculum, course, or lesson. The approach and organization is stored in the Transaction and Enterprise Knowledge Bases. The content is stored in the Domain Knowledge Base.
- c. The Transaction Authoring System contains a library of reusable instructional programs, or transaction shells, for the delivery of instruction. These programs contain generalized instructional algorithms, each appropriate for teaching a certain type of content, but do not contain any content. A transaction shell is a piece of computer code which, when executed, causes a given transaction to take place. Each shell incorporates a number of parameters, configurable by the author, which control the functioning of the shell during course delivery. Each shell knows what knowledge it must have in order to execute its interaction with the learner. It is able to query the domain knowledge base to find the required knowledge, and thus be able to instantiate its knowledge slots. If the domain knowledge base does not contain the necessary knowledge, the transaction shell can direct the SME/ID to supply the required content. Once a transaction has been selected or prescribed, it must then be configured and authored. Configuration involves setting the parameters, modifying the strategy, and attaching the content. Authoring involves attaching domain specific instructional materials to the instructional structure set up by the transaction. Each transaction shell has default values for each of its parameters, including its strategy elements.

- d. The Instruction Delivery System. Authoring decisions made at design time are based on the designer's best estimate of the student population. During the delivery of instruction, information about the student -- aptitude, specific goals, motivation, familiarity, and other factors, as well as the student's expressed preferences -- may be taken into account to modify those decisions. The on-line delivery advisor has access to the domain knowledge base and the configurations. In addition, it maintains a student model that contains information about the student. Using the information gathered about the student, the advisor adjusts design decisions to customize the instruction to more adequately meet the characteristics of the student. The advisor also can engage in a mixed-initiative dialogue with the student which allows the student to participate in this decision-making.

The basic unit of instruction delivery is the interaction with the student. Transactions are comprised of interactions that are characterized by certain interaction modes, methods of interaction with the student. Interaction modes assume different values on the form of the interaction (expository or inquisitory) and the degree of learner control involved (learner control or system control). Five interaction modes have been identified: overview, presentation, practice, generality assessment, and instance assessment. Interaction strategy is the combination and sequence of interaction modes available to the student. There are seven interaction strategies: overview, familiarity, basic, mastery, basic remediation, mastery remediation, summary, and assessment.

- e. The Evaluation System Once the instructor designs a transaction (TRX), the student, or the instructor playing the role of the student, goes through the TRX. The interactions of the transaction are stored in the Session Knowledge Base. The Evaluation System analyzes the Session KB and either proposes changes or makes changes in the transaction shell (TRXS).

Instead of being part of the Delivery System, the Evaluation System could be part of this and make on-line changes to the TRXSs.

- f. The Courseware Generation System Delivery of instruction to a large number of students may not be a function of the AIDA system. Instruction delivery is more likely to be accomplished via a multi-terminal, multi-media system optimized for instruction delivery, such as ISS, WISE, QUEST, PILOT, TENCORE, etc. The Courseware Generation System analyzes the configured and authored transactions and generates courseware in a format that can be used by other instruction delivery systems. XAIDA will, however, have a

student mode which will enable the SME/ID to view the instruction sequence as seen by the student in order to evaluate it.

3.2.2 Knowledge Base Interrelationships

3.2.2.1 Domain Knowledge Base (DOMAIN.KB)

DOMAIN.KB Functional Overview. The function of the Domain Knowledge Base (DOMAIN.KB) is to represent all the knowledge required for instruction leading to [the] acquisition of an integrated human performance, or enterprise. This knowledge includes the cognitive model(s) of the domain knowledge as well as the instructional material associated with that knowledge.

The three kinds of domain knowledge are entities, activities, and processes. For each of these kinds of domain knowledge, the DOMAIN.KB must specify the following:

- a. Attributes: Attributes represent the characteristics of an entity, activity, or process such as its name and function. In the DOMAIN.KB, attributes also represent instructional material such as figures and explanations.
- b. Components: Components represent the constituents of a frame. For an entity, the components would be parts of the entity; for an activity, steps; and for a process, events and causes.
- c. Abstractions: Abstractions represent the class/sub-class hierarchy into which an entity, activity, or process may be classified.
- d. Associations: Associations represent other associations (relationships) that an entity, activity, or process can have with other entities, activities, and processes. These include relations such as (1) analogy for, (2) alternate for, (3) uses, (4) involves, and (5) applies.

DOMAIN.KB Design and Relationships. The DOMAIN.KB is self-contained and does not reference other knowledge bases. It is comprised of the following frames:

Domain Knowledge Frame. The basic unit of the Domain Knowledge Base is the Domain Knowledge Frame. A Domain Knowledge Frame describes content, which can be an entity, activity or process. It specifies the following:

- a. Kind: whether the content specified by the frame is an entity, activity or process.
- b. Attributes: the Attributes Frame which specifies the attributes of the entity, activity, or process.

- c. Components: the Components Frame which specifies the component hierarchy to which the entity, activity, or process belongs.
- d. Abstractions: list of Abstraction Frames which specify the class/sub-class hierarchy to which the entity, activity, or process belongs.
- e. Associations: list of Association Frames which specify the associations (relations) of the entity, activity or process to other entities, activities and processes.

Entity Attributes Frame. This frame specifies the attributes of entities. This frame is also used to specify the parts of entities. Example attributes are Name, Description, Location, and Figure.

Activity Attributes Frame. This frame specifies the attributes of Activities. This frame is also used to specify the steps of activities. Example attributes are Name, Description, Event, and Sequence.

Process Attributes Frame. This frame specifies the attributes of Processes. This frame is also used to specify the states and causes of processes. Example attributes are Name, Description, Inputs, Outputs, Transformations, Events, and Timing.

Components Frame. This frame specifies the component hierarchies of entities, activities, and processes. The Components Frame specifies the following:

- a. Knowledge: the Domain Knowledge Frame which specifies the component.
- b. Parent: the Components Frame which specifies the entity, activity, or process of which the component is a subcomponent.
- c. Children: the list of components frames that specify the component Domain Knowledge Frames.

The relationship between Component Frames and Domain Knowledge Frames is illustrated in Figure D-2.

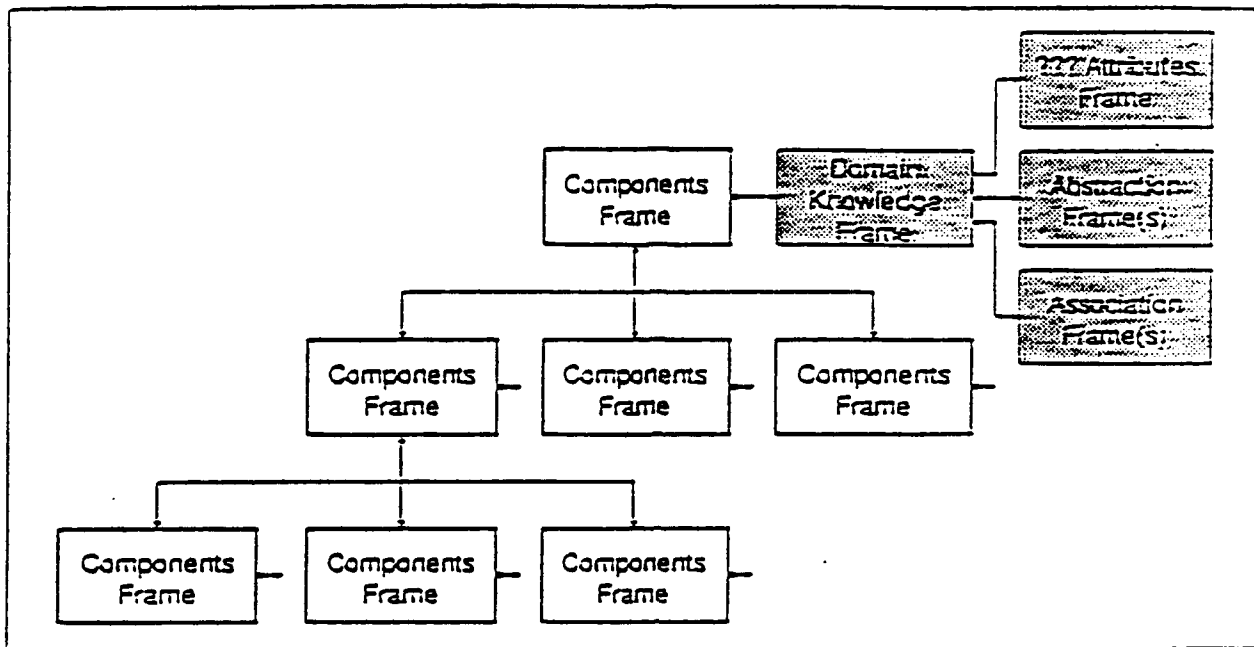


Figure D-2: Component hierarchy diagram.

Abstraction Frame. Abstraction Frames specify the class/sub-class hierarchies of entities, activities, and processes. The Abstraction Frame specifies the following:

- a. Name: the name of the class/sub-class.
- b. Description: a description of the class/sub-class.
- c. Attributes: the (properties) that characterize the class/sub-class.
- d. Parent: the Abstraction Frame which specifies the super-class of the class.
- e. Children: list of Abstraction Frames which specifies the sub-classes of the class.
- f. Members: list of Domain Knowledge Frames that are members of the class/sub-class.

The Abstraction Frame has a second use. It can also be used to represent "Collections" which are organizations of entities, activities, and processes that have no relation other than grouping. (See SS 3.2.1.1)*

Association Frame. The following associations/relations are described in section 3.3.1.7.

```
"A" USES "X", "Y", "Z"
"A" INVOLVES (REQUIRES) "X", "Y", "Z"
"A" APPLIES "X", "Y", "Z"
"A" ANALOGY_FOR "B"
"A" ALTERNATIVE_FOR "B"
```

The following generalization can be used to assert USES, INVOLVES, and APPLIES associations:

OBJECT ASSOCIATION LIST_OF_ASSOCIATED_OBJECTS

For example:

```
"A" USES ["X", "Y", "Z"]
"A" INVOLVES ["X", "Y", "Z"]
"A" APPLIES ["X", "Y", "Z"]
```

The last ANALOGY_FOR and ALTERNATIVE_FOR associations are used to assert that a set of objects are interchangeable analogies or alternatives for one another. The following generalization can be used to assert these "set" relations:

ASSOCIATION LIST_OF_ASSOCIATED_OBJECTS

For example:

```
ANALOGY_FOR ["A", "B", "C"]
ALTERNATIVE_FOR ["A", "B", "C"]
```

This is far more efficient than asserting each association separately as:

```
"A" ANALOGY_FOR "B"
"A" ANALOGY_FOR "C"
"B" ANALOGY_FOR "C"
"A" ALTERNATIVE_FOR "B"
"A" ALTERNATIVE_FOR "C"
"B" ALTERNATIVE_FOR "C"
```

* Section 3.2.1.1 of the System Specification for the Experimental Advanced Instructional Design Advisor (XAIDA), CDRL Seq. No. 31, Contract No. F33615-88-C-0003, Task Order No. 0006.

However, while these two syntaxes for asserted association are generalized and powerful, a third, more generalized syntax is implemented in the Association Frame. It specifies the following:

- a. Association: the or label or the association (ANALOGY_FOR, USES, ...).
- b. Objects: list of objects.
- c. Associated Objects: list of associated objects.

In other words, this structure implements the following syntax:

ASSOCIATION LIST_OF_OBJECTS LIST_OF_ASSOCIATED_OBJECTS

Elaborated Frame Network. (See SS 3.2.1.1) Together, the frames in the DOMAIN.KB are organized into an "Elaborated Frame Network" as illustrated in Figure D-3.

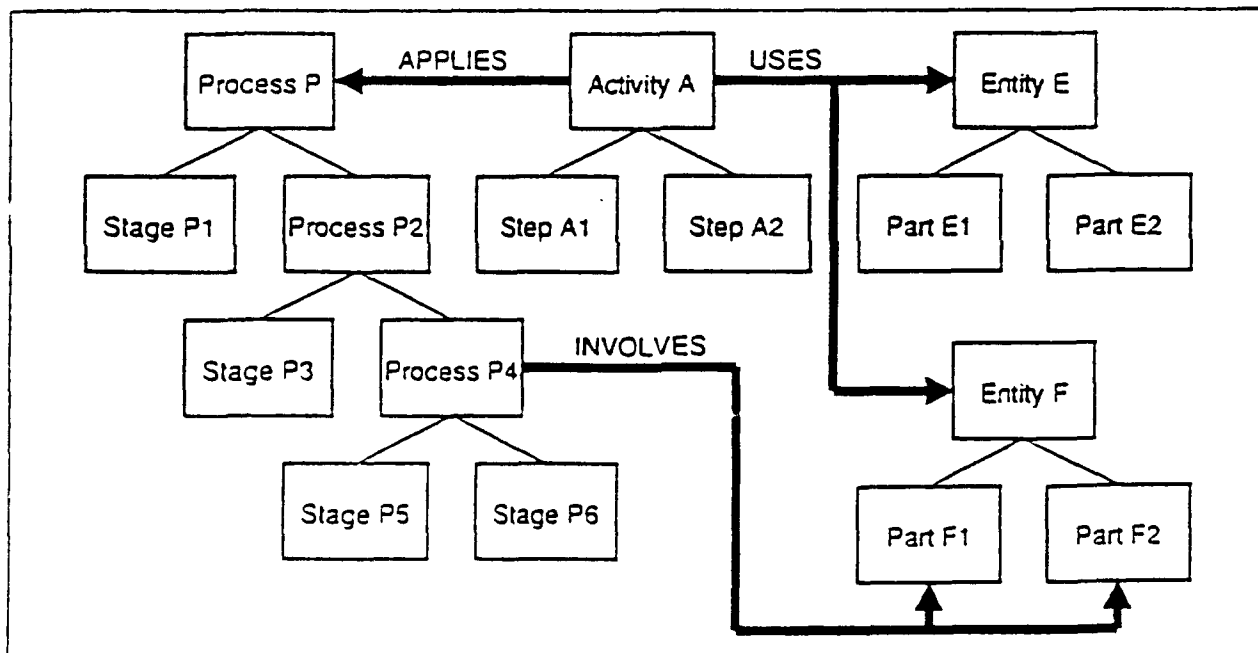


Figure D-3: Elaborated Frame Network (EFN) example diagram.

DOMAIN.KB and System Relationships. Here we describe how the Domain KB is used by the system. TRXSs are driven by the Domain KB, including both the domain knowledge and the parameters. A TRXS is a piece of code that says: For a given piece of knowledge, perform this kind of transaction (TRX). The parameters specify some of the characteristics of that TRX. The domain knowledge is the content of the TRX. So the Domain KB drives the TRXS. For example, the Naming TRXS only knows what to do if it is given a list of components drawn from the Domain KB.

As shown in Figure D-1, the SME uses the Knowledge Acquisition System (KAS) to fill the Domain KB. The Domain KB is analyzed by the Strategy Analysis System (SAS). The Domain KB, along with the other KBs, i.e., the TRX KB, the Enterprise KB, etc., drive the TRXSS.

3.2.2.2 Transaction Knowledge Base (TRANSACTION.KB)

TRANSACTION.KB Functional Overview. The function of the Transaction Knowledge Base (TRANSACTION.KB) is to represent descriptions of instructional transactions, where a transaction (TRX) is defined to be a particular interaction with a student. For each transaction, the TRANSACTION.KB must specify the following:

- a. Transaction Shell (TRXS): The TRXS is a piece of code which when executed causes a particular transaction to take place.
- b. Focus: The focus is the domain knowledge required to perform a particular interaction.
- c. TRXS Parameters: TRXS parameters configure the operation of TRXS. The parameters are Content, Coverage, Guidance Level, Guidance Type, View, Vertical Sequence, Temporal Sequence, Trials, Mastery Level, Response Mode, Feedback, Replacement, Items, Timeout, Item Order, Modes, Strategy, Strategic Control, and Tactical Control.

TRANSACTION.KB Design and Relationships. The Transaction Knowledge Base is comprised of Transaction Frames and TRXS Parameter Frames which describe a transaction by specifying (1) the TRXS which will perform the transaction, (2) the focus knowledge of the transaction, and (3) the parameters specifying the performance characteristics of the particular transaction.

Transaction Frame. The Transaction Frame specifies the following (see SS 3.2.1.3):

- a. Name: The name of transaction.
- b. Description: The description of the transaction.
- c. TRXS: The transaction shell (e.g., Denote, Evaluate, Execute, Design, Interpret)
- d. Performance Level: (Denote, Perform for Execute Kind of TRXS) or (Denote, Explain, Predict for Interpret Kind of TRXS)
- e. Focus: (See SS 3.2.1.3)

- f. Parameters: The parameters of a transaction including Content, Coverage, Guidance Level, Guidance Type, View, Vertical Sequence, Temporal Sequence, Trials, Mastery Level, Response Mode, Feedback, Replacement, Items, Timeout, Item Order, Modes, Strategy, Strategic Control, and Tactical Control

TRXS Parameters Frame. The TRXS Parameters Frame specifies the following parameters:

Content, Coverage, Guidance Level, Guidance Type, View, Vertical Sequence, Temporal Sequence, Trials, Mastery Level, Response Mode, Feedback, Replacement, Items, Timeout, Item Order, Modes, Strategy, Strategic Control, and Tactical Control.

TRANSACTION.KB and System Relationships. The information in the TRX KB parameterizes the TRXS.

In Figure D-1, it is called the TRX Authoring System. It might also be called the TRX Configuration System. The TRX Authoring System parameterizes the TRXSs, and that information is stored in the TRX KB. The information in the TRX.KB then drives the TRXSs, resulting in the instruction.

3.2.2.3 Enterprise Knowledge Base (ENTERPRISE.KB)

ENTERPRISE.KB Functional Overview. The function of the Enterprise Knowledge Base (ENTERPRISE.KB) is to represent Enterprise Transactions.

Enterprise Transactions. An enterprise is a complex human performance that requires an integrated set of knowledge and skills. The goal of instruction is the acquisition by the learner of one or more enterprises. The primary transaction shells, described in Section 3.2.1.2 of the System Specification, facilitate the acquisition of the knowledge and skills which comprise enterprises, but by themselves cannot accomplish their integration. This integration must be accomplished by enterprise transactions.

Transaction Family. The transactions necessary to acquire all the knowledge and skills associated with a given enterprise comprise a transaction family. In maintenance, the six enterprises are equipment operation, equipment calibration and adjustment, equipment testing, access and disassembly, equipment repair, and troubleshooting. The type and sequence of interactions necessary to acquire each of these complex enterprises are different.

Transaction Manager (TRXM). A transaction manager is required for each enterprise. The transaction manager is a program that calls and sequences the primary transactions identified as necessary for a particular course or curriculum.

Equipment Model Transaction Family. In addition to a transaction family for each of the 6 maintenance training enterprises, an equipment model transaction family is a component of each of the other transaction families. The equipment model transaction family is not a "stand-alone" family, but is a component of the transaction family required for each of the other 6 maintenance training enterprises. Two classes of transactions are represented in the equipment model TRX family: identify to teach physical & conceptual structure; and interpret - to teach device functioning, device configuration, fault recognition, and prediction. The equipment model TRX family also provides for the integration of the learning facilitated by the primary transactions, ideally as a performance or simulation of an activity that is representative of the real-world performance of the enterprise.

A course is a set of enterprise transactions and their supporting families of primary transactions. A course organization is a nesting and/or ordering of the enterprise transactions.

ENTERPRISE.KB Design. The ENTERPRISE.KB is comprised of a set of Enterprise Frames which specifies a network of enterprise transactions.

ENTERPRISE.KB and System Relationships. The network of enterprises describes a course. As shown in Figure D-1, the enterprise network or knowledge base is the input to the transaction manager and instruction delivery system. The transaction manager sorts through the enterprise network and enterprise frames, and points to transactions to be executed in the course. In short, the Enterprise KB is the input to the transaction manager and instruction delivery system.

3.2.2.4 Student Knowledge Base (STUDENT.KB)

STUDENT.KB Functional Overview. Using the Knowledge Acquisition/Representation System (KARS), AIDA gathers information about the student, as shown in Figure D-1. Then the Strategy Analysis System (SAS) analyzes the student, the instructional environment and the task, and builds an enterprise network or knowledge base.

XAIDA will be designed with basic information about the students and the instructional environment already in the system. The instructional designer will be given the capabilities to be acquired by the student and will enter them into XAIDA. XAIDA will then select and configure transaction shells appropriate to the specified capabilities. The SME/ID will then be prompted to enter any needed content knowledge to complete a frame appropriate to the particular knowledge type.

Because we are assuming a fixed instructional environment (small class, computer-based instruction, located at a TTC) and students who are readers and reasonably motivated, information about the

students and instruction environment will be hard-coded into XAIDA for the time being.

STUDENT.KB Design and Relationships. The Student Knowledge Base is comprised of Student Profile Frames.

Student Profile Frame. A Student Profile Frame specifies (1) the name and description of the set of students being profiled, and (2) the Control, Motivation, Familiarity, Ability, and Roles profiles of those students.

STUDENT.KB and System Relationships. Using the Knowledge Acquisition/Representation System (KARS), AIDA gathers information about the student, as shown in Figure D-1. Then the Strategy Analysis System (SAS) analyzes the student, the instructional environment and the task, and builds an enterprise network or knowledge base.

3.2.2.5 Environment Knowledge Base (ENV.KB)

(See explanation in Section 3.2.2.4.)

ENV.KB Functional Overview. (See explanation in Section 3.2.2.4)

ENV.KB Design and Relationships. The Environment Knowledge Base is comprised of Environment Profile Frames.

Environment Profile Frame. An Environment Profile Frame specifies (1) the name and description of the environment being profiled, and (2) Delivery Medium, Location, Schedule, Grouping, and Instructor profiles of that environment.

ENV.KB and System Relationships. Using the Knowledge Acquisition/Representation System (KARS), the SME/ID AIDA gathers information about the instruction environment (see Figure D-1). Then the Strategy Analysis System (SAS) analyzes the instruction environment, the student and the task, and builds an enterprise network or knowledge base.

3.2.2.6 Task Knowledge Base (TASK.KB)

TASK.KB Functional Overview.

Task Analysis in AIDA. The first step in the design of any instruction is a task analysis to determine what should be taught. A behavioral analysis is not sufficient; a cognitive analysis must be performed to take into account the cognitive processes involved in learning and performance.

In AIDA Phase 2, Halff identified three types of cognitive structures important to the maintenance enterprise: (1) the execution of procedures, (2) a mental model of the equipment, and (3) fault isolation skills. AIDA will identify the information and skills that must be imparted to the student to support the

acquisition of these three cognitive structures. (XAIDA will address only the areas of procedures and mental models, omitting fault isolation.)

To do a task analysis, AIDA will employ the GOMS (Goals, Operators, Methods, and Selection Rules) method, in which the tasks to be accomplished are broken down into a series of goals and subgoals until a level is reached in which the subgoal can be achieved by either a primitive level motor or mental act.

Kieras has prepared a detailed guide for doing a GOMS task analysis. He has also defined a language called Natural GOMS Language (NGOMSL). (For a more detailed discussion of the cognitive analysis of procedures, see the System/Segment Specification, Section 3.2.2.6.)

The GOMS task analysis yields three hierarchically arranged representations: (1) The steps to be followed in carrying out activities for operating, calibrating, troubleshooting, or repairing the equipment starting with the highest level goals and methods. These are successively decomposed to lower-level subgoals and methods. (2) The device components that need to be included in the representation of the device structure, as well as the declarative knowledge that needs to be conveyed about them, i.e., function, location, name, etc. (3) The causal and declarative knowledge necessary to execute the procedures, support inferences necessary for constructing a mental model of the equipment, and define the attributes and rules of objects, etc.

The results of the GOMS task analysis will be implemented in a working simulation. The device simulation will contain a graphic representation of the device structure and qualitative simulations of its functioning. Authoring the simulation will begin with a temporary sketch derived from the cognitive analysis, particularly the mental model analysis which entails inter-relating the GOMS analysis, the explanation hierarchy, and the hierarchical device structure decomposition. However, the construction of the simulation will be done from the bottom up, starting with the lowest level of the device hierarchy. The lowest level objects - the bottom items in the device structure analysis - correspond to the objects manipulated by the lowest level operators in the GOMS model.

The behaviors of objects are defined by attribute handles and rules drawn from the explanation hierarchy. Once the basic simulation is complete, procedures which are carried out on the device are authored by carrying out a sequence of actions which correspond to the activities which accomplish the goals in the GOMS analysis. The individual actions correspond to the operators. The function or purpose, i.e. goals, of the procedure will be presented in dialogue windows.

Representing The Task Analysis in AIDA. The device knowledge necessary to carry out the simulation can be represented using Anderson's Penultimate Production System (PUPS). In PUPS, the declarative knowledge necessary for compiling the procedures which model the task performance is represented in schema-based structures compatible with the AIDA representation. The PUPS schema include slots for the function of the entity, a form slot for the physical appearance of the entity, and a precondition slot which states the preconditions necessary for the function to be achieved.

In compiling the productions which are the basis of procedural knowledge, the function slot maps to the goal to be achieved which will require knowledge of the entity represented; the preconditions slot maps onto the condition of the condition-action pair in a production. The form slot holds the form of the current action to be carried out, such as a particular LISP function.

As explained in Sections 3.2.2.1 and 3.3.1.3, the AIDA Activity frame has paths or sequences of actions. This frame can also have slots for the function, the operators, and the outcome. Later, the values for these slots may be automatically generated from a NGOMSL analysis and generate a production rule-based simulation.

Task Analysis in XAIDA. The GOMS task analysis approach will be used in the task analysis module of XAIDA which includes (1) a GOMS analysis for the procedural aspects of the task, (2) a mental model analysis to develop an explanation hierarchy, and (3) a decomposition of device structure and function, and relating them to the GOMS analysis.

Shells, paper-based or computer-based, will aid in the task analysis. A shell will guide the instructional designer in doing a GOMS analysis of a particular task, using either the documentation at hand or the knowledge of an SME. The shell will be based on Kieras' GOMS manual, and may be developed in NGOMSL. Kieras' rules of thumb will be implemented in a knowledge-based shell to give guidance to the SME/ID.

In XAIDA, the task analysis module will do much of the bookkeeping necessary for a GOMS analysis, creating a list of methods and information that must be either known or taught, such as their location, etc. (Later, in AIDA, a more sophisticated shell will automatically map the results of the analysis into the KARS). XAIDA will employ a paper guide for what should be hand entered into the KARS. Similar shells will be created for the explanation hierarchy, and the device structure and function knowledge.

XAIDA is focused on teaching procedures for electronics maintenance. An enterprise analysis pertinent to that domain will be customized for XAIDA, as will an elaborated frame network shell pertinent to procedures used in electronics maintenance.

Nevertheless, to ensure future compatibility with AIDA, the KARS in XAIDA will be developed around the concepts of cognitive task analysis, described in the System/Segment Specification, Section 3.2.2.6.

TASK.KB Design. The Task Knowledge Base is comprised of Task Frames.

Task Frame. The Task Frame specifies (1) the name and description of the task to be taught, and (2) the activities that describe the task.

TASK.KB and System Relationships. Using the Knowledge Acquisition/Representation System (KARS), AIDA gathers information about the task to be learned, as shown in Figure D-1. Then the Strategy Analysis System (SAS) analyzes the task, the student, and the instruction environment, and builds an enterprise network or knowledge base (ENTERPRISE.KB).

3.2.2.7 Session Knowledge Base (SESSION.KB)

SESSION.KB Functional Overview. The performance of the student in each instructional session will be captured and stored in the SESSION.KB. The data stored in the SESSION.KB will be used in (1) student/course management, and (2) formative evaluation during course development. For example, the student's place in the course at the end of the session will be used to determine the starting point in the next session. Other data will be analyzed in the EVALUATION.KB so that the author can make adjustments in the design of the course during course development. For example, the author may adjust the limits on learning vs. system control, or delete nondiscriminating test items, etc.

SESSION.KB and System Relationships. Data in the SESSION.KB are analyzed by the Evaluation System, leading to changes in the Transaction Shells (TRXSs).

3.3 Knowledge Base Design

The following paragraphs describe representative frames in the AIDA knowledge bases by listing for each knowledge base the constituent frames, slots, and facets.

3.3.1 **KB: Domain Knowledge Base**

3.3.1.1 FRAME: Domain Knowledge Frame (see DBDD 3.2.2.1)

FRAME: Domain Knowledge Frame
 SLOT: Kind
 SLOT: Attributes

SLOT: Components
SLOT: Abstractions
SLOT: Associations

SLOT: Kind. The domain knowledge frame is a generalized structure that can specify the three kinds of domain knowledge: entity, activity, and process. This slot specifies the kind of domain knowledge.

There are three possible values corresponding to the three kinds of domain knowledge:

- a. ENTITY: A thing. For example, a device, person, creature, place or symbol.
- b. ACTIVITY: A group of related actions to be performed by the learner.
- c. PROCESS: A group of related actions entirely external to the learner.

SLOT: Attributes. This slot points to an Entity Attributes Frame, Activity Attributes Frame, or Process Attributes Frame which specifies the attributes of the Entity, Activity or Process.

SLOT: Components. This slot points to a Components Frame which specifies the component hierarchy to which the Entity, Activity, or Process belongs. The Components Frame specifies the parent and sub-components of the Entity, Activity or Process. Refer to the description of the Components Frame (Section 3.2.2.1) for more detail.

SLOT: Abstractions. This slot lists the Abstraction Frames that specify the class/subclass hierarchy(s) to which the Entity, Activity, or Process belongs. Refer to the description of the Abstraction Frame (Section 3.2.2.1) for more detail.

SLOT: Associations. This slot lists the Association Frames that specify the associations (relations) of the Entity, Activity, or Process to other Entities, Activities, and Processes. Refer to the description of the Association Frame (see 3.2.2.1) for more detail.

3.3.1.2 FRAME: Entity Attributes Frame

FRAME: Entity Attributes Frame
SLOT: Name
SLOT: Description
SLOT: Location
SLOT: Figure
FACET: Format
FACET: File

SLOT: Name. This slot specifies the name of the entity or part.

SLOT: Description. This slot describes the entity or part.

SLOT: Location. This slot is applicable only to components. It specifies the relative location of the component on the figure representing the parent entity.

SLOT: Figure. This slot specifies the format and source of the view (if any) associated with the entity. Note that primary (top level) entities must have figures, but components may or may not have figures.

FACET: Format. This facet specifies the graphics format of the view of the entity. (See Section 3.1.2 (e))

FACET: File. This facet specifies the name of the file (or some other pointer mechanism) containing the figure of the entity or component.

3.3.1.3 FRAME: Activity Attributes Frame

FRAME: Activity Attributes Frame

SLOT: Name
SLOT: Description
SLOT: Event
SLOT: Sequence

SLOT: Name. This slot specifies the name of the Activity, Step or Action.

SLOT: Description. This slot describes the Activity, Step or Action.

SLOT: Event. This slot applies only when the frame is describing a step of an activity. It specifies the condition when this step can be performed.

SLOT: Sequence. This slot specifies the sequence, including loops and conditions, of the Activity's component steps or actions.

3.3.1.4 FRAME: Process Attributes Frame

FRAME: Process Attributes Frame

SLOT: Name
SLOT: Description
SLOT: Inputs
SLOT: Outputs
SLOT: Transformation
SLOT: States
SLOT: Timing

SLOT: Name. This slot specifies the name of the process or stage.

SLOT: Description. This slot describes the process or stage.

SLOT: Inputs. This slot specifies the inputs to the process or stage. This should include the name of the input, unit of measure and range.

SLOT: Outputs. This slot specifies the outputs to the process or stage. This should include the name of the output, unit of measure and range.

SLOT: Transformations. This slot specifies the internal equations that transform inputs to outputs.

SLOT: States. This slot specifies the possible states of the process or stage. Each state has a name and is described by the range of input and/or output values that describe the state.

SLOT: Timing. This slot specifies the timing of the process or stage. It is a special "input" with complex transformations.

3.3.1.5 FRAME: Components Frame

FRAME: Components Frame

SLOT: Knowledge

SLOT: Parent

SLOT: Children

SLOT: Knowledge. This slot specifies the knowledge frame that describes the Entity, Activity or Process or its components.

SLOT: Parent. This slot points to the parent component frame which specifies the parent knowledge.

SLOT: Children. This slot lists the children component frames.

3.3.1.6 FRAME: Abstraction Frame

FRAME: Abstraction Frame

SLOT: Name

SLOT: Description

SLOT: Properties

SLOT: Parent

SLOT: Children

SLOT: Members

SLOT: Name. This slot specifies the name of the class/sub-class.

SLOT: Description. This slot describes the class/sub-class.

SLOT: Properties. This slot specifies the properties that characterize the class/sub-class.

SLOT: Parent. This slot points to the Abstraction Frame which specifies the superclass of the class.

SLOT: Children. This slot lists the Abstraction Frames which specify the sub-classes of the class.

SLOT: Members. This slot lists the Domain Knowledge Frames that are members of the class/sub-class.

3.3.1.7 FRAME: Association Frame

FRAME: Association Frame
 SLOT: Association
 SLOT: Objects
 SLOT: Associated_Objects

SLOT: Association. This slot specifies the association by name.

SLOT: Objects. This slot specifies the object or objects of the association.

SLOT: Associated Objects. This slot specifies the associated object or objects.

3.3.2 **KB: Transaction Knowledge Base**

3.3.2.1 FRAME: Transaction Frame

FRAME: Transaction Frame
 SLOT: Name
 SLOT: Description
 SLOT: TRXS
 SLOT: Performance Level
 SLOT: Focus
 SLOT: Content
 SLOT: Coverage
 SLOT: Guidance Level
 SLOT: Guidance Type
 SLOT: View
 SLOT: Vertical Sequence
 SLOT: Temporal Sequence
 SLOT: Trials
 SLOT: Mastery Level
 SLOT: Response Mode
 SLOT: Feedback
 SLOT: Replacement Items
 SLOT: Items
 SLOT: Timeout
 SLOT: Item Order

SLOT: Modes
SLOT: Strategy
SLOT: Strategic Control
SLOT: Tactical Control

SLOT: Name. This slot specifies the name of the transaction as supplied by the SME/ID.

SLOT: Description. This slot describes the transaction to the user.

SLOT: TRXS. This slot specifies the transaction shell which will be used to perform the transaction. There will be a TRXS for each of the twelve primary transactions. They are:

- a. Identify: An identify transaction requires either an instance or class entity frame. It enables the student to acquire the names, functions, properties, and relative location of all the parts which comprise an entity. The student knows what it is.
- b. Execute: An execute transaction requires either an instance or class activity frame. It enables the student to acquire the steps of an activity. The student knows how and is able to [perform] the activity.
- c. Interpret: The interpret transaction requires either an instance or class process frame. It enables the student to acquire the events and causes in a process. This means that the student knows why it works and can explain the events which lead to a given consequence, or can predict the consequence from a series of events.
- d. Classify/Decide: A classify/decide transaction requires a superclass frame with two or more subordinate class frames each of which have two or more instance frames. These frames can be entity, activity, or process frames. It enables the student to acquire the ability to sort or classify instances as to class membership. It enables the student to know when to select one alternative from another. Concept identification is an example. Deciding among alternative activities to accomplish some goal is an example. Editing (selecting the appropriate usage) is an example.
- e. Judge: A judge transaction requires an abstraction frame with two or more subordinate instance frames. These frames can be entity, activity, or process frames. It enables the student to acquire the ability to order the instances of a given class on the basis of some dimension (criterion). The dimensions can be any attribute or combination of attributes. Judging the performance of others as they perform an activity is an example. Ordering a set of objects is an example.

- f. Generalize: A generalize transaction requires a superclass frame with two or more subordinate class frames each of which have two or more instance frames. These frames can be entity, activity, or process frames. Generalization transactions enable the student to acquire the ability to combine instances of two or more classes into a more general class. Generalization is the inverse of classification.
- g. Transfer: A transfer transaction requires a superclass frame and one or more class frames. These frames can be entity, activity, or process frames. It enables the student to acquire an abstraction model, that is, a generalized set of steps for an activity, or a generalized set of events for a process, and to apply this abstraction model to a previously unencountered class or instance of the activity or process.
- h. Propagate: A propagate transaction requires two or more associated frames. The most common relations between knowledge frames - uses, requires, applies - all involve propagation. A propagation transaction makes a deliberate effort to facilitate the student's integration of information from two or more associated knowledge frames. One of the most important propagation associations is the link between an application activity and tool activity; another is the link between a method activity and a process. Propagation enables the student to acquire one set of skills in the context of another set of skills. While learning an application activity the student can simultaneously learn the tool for doing the application. Or while learning a tool, the student can simultaneously learn application activities for the tool. Or while learning a process, the student can simultaneously learn a method activity for studying or observing the process. Or while learning a method activity, the student can simultaneously learn the process for which the method was devised.
- i. Analogize: An analogize transaction requires two or more knowledge frames linked by the relation "analogy for." It enables the student to acquire the steps from one activity by likening it to an analogous activity; or to acquire the events in one process by likening it to an analogous process or activity.
- j. Substitute: A substitute transaction requires two or more knowledge frames linked by the relation "alternative for." It enables the student to learn an alternative activity or process by comparison, elaboration, or extension of a previously learned activity or process. It also enables the student to acquire alternative ways to accomplish a given activity or to explain a given process.

k. Design

l. Discover

XAIDA will contain TRXSSs for the first four transactions:
Identify, Execute, Interpret and Decide.

SLOT: Performance Level. Performance Level is a parameter to Execute and Interpret transaction shells. For Execute, the values may be either Denote or Perform; for Interpret, values are either Denote, Explain, or Predict.

SLOT: Focus. Focus is a pointer to a knowledge frame in the EFN.

SLOT: Content. Indicates how much of the component hierarchy is available for instruction. Takes one of the following (default is ALL):

- a. All: entire hierarchy.
- b. List: a list of frames (subset of the hierarchy) which are the content.

SLOT: Coverage. This slot indicates how much of the component hierarchy for the focus is to be instructed. Takes one of the following (default is Focus):

- a. All: entire hierarchy.
- b. Focus: the focus, its superpart (the single frame directly above the focus in the hierarchy), and its first level of subparts.
- c. Levels: requires an additional integer argument, which indicates how many levels of subparts of the focus are instructed.
- d. Exemplar, label: the focus and a single, specified subpart are instructed.
- e. Random Exemplar: the focus and single, randomly selected subpart are instructed.

SLOT: Guidance Level. Sets the level of guidance to the learner. Takes one of the following (default is Full):

- a. None: no guidance;
- b. OnDemand: guidance presented on learner request only;
- c. Full: guidance at all times;
- d. Faded: begin with full guidance, fade to OnDemand by end of transaction.

SLOT: Guidance Type. Takes one of two values (default is Verbose):

- a. Concise: guidance interactions are short and to the point;
- b. Verbose: guidance interactions are detailed and complete.

SLOT: View. Representation of the subject matter. Takes one or more of the following (default is Structural):

- a. Structural: displays the component relation in terms of the knowledge structures, in tree format;
- b. Physical: displays an author-supplied graphic or illustration of the object whose parts are being instructed, representing the physical appearance of the object;
- c. Functional: displays an author-supplied graphic or illustration of the object whose parts are being instructed, representing the functional appearance of the object.

SLOT: Vertical Sequence. Order of introduction of the parts. Takes one of the following values (default is TopDownBreadth):

- a. TopDownBreadth: ordering is from the highest superpart to the lowest level, breadth first (an entire level is introduced before any components of the next level are introduced);
- b. TopDownDepth: ordering is from the highest superpart to the lowest level, depth first;
- c. BottomUp: ordering is from the lowest subpart level to the highest, breadth first.

SLOT: Temporal Sequence. Within the vertical sequence, is the order of introduction of the parts on the same level. The default is LeftRight:

- a. LeftRight: ordering is accordingly to the representation in EFN, from left to right;
- b. LowHigh,attribute label: ordering is according to the ranking of a named attribute, from low to high;
- c. HighLow,attribute label: ordering is according to the ranking of a named attribute, from high to low.

SLOT: Trials. The number of times to sequence through the set of parts. Takes a positive integer value. Default is 1.

SLOT: Mastery Level. The percent correct for mastery. Takes a value between 0 and 100. Default is 80%.

SLOT: Response Mode. Type of response performance required of the learner. Takes one of two values, either Recognition or Recall. Default is Recognition.

SLOT: Feedback. Timing of feedback. Takes one of the following values (default is PrePractice):

- a. None: no feedback is given;
- b. PostResponse: corrective feedback is given immediately after a response;
- c. PrePractice: corrective feedback is given just before the next opportunity to practice.

SLOT: Replacement Items. Whenever sampling is used in the transaction, this parameter controls whether a new sample may or may not include items from previous samples. The default is With.

- a. With: a new sample may include items previously used;
- b. Without: new samples are distinct from previous samples.

SLOT: Items. Whenever a pool of items is practiced or tested, this parameter sets the maximum size of the pool. It takes a positive integer value. The default is 3.

SLOT: Timeout. The amount of time to wait for a user response before timing out. If the user response involves typing rather than pointing, the timeout occurs after a base interval, plus a fraction of the base interval multiplied by a number derived from the length of the expected response. If the user response is pointing only, the timeout occurs after the base interval. The value of the parameter is the base interval, a positive integer. The default is 3 (seconds).

SLOT: Item Order. Whenever a pool of items is practiced or tested more than once, this parameter controls whether the ordering is the same or different. The default is Random.

- a. Random: the ordering is random;
- b. Same: the ordering is fixed.

SLOT: Modes. The mode is the method of interaction with the student. One or more of the following may be selected (default is Overview):

- a. Overview: presents the knowledge structure from the knowledge base in the Structural view;

- b. Presentation: presents an author-supplied graphic in either the Physical or the Functional view, and demonstrates the parts to the learner;
- c. Practice: provides practice for the learner using the author-supplied graphic, in either the Physical or Functional view;
- d. Instance Assessment: tests the student's mastery of the material, using the author-supplied graphic, in either the Physical or Functional view.

SLOT: Strategy. Strategy is defined as a sequence of modes. Because modes are fully determined by strategy, the arguments to the Mode parameter are ignored unless the Strategy parameter is None (the default):

- a. Overview: the Overview mode;
- b. Familiarity: the Overview mode followed by the Presentation mode;
- c. Basic: Overview plus Presentation plus Practice modes;
- d. Mastery: Overview plus Presentation plus Instance Assessment;
- e. Basic Remediation: Instance Assessment, followed by Basic Strategy to remediate errors;
- f. Mastery Remediation: Instance Assessment, followed by Mastery Strategy to remediate errors;
- g. Assessment: Instance Assessment mode;
- h. Summary: Overview with Coverage set to Focus, plus Presentation followed by Instance Assessment, with Coverage set to Random Exemplar for both;
- i. None: No strategy.

SLOT: Strategic Control. Determines the level of control granted the learner over the selection of strategy, mode, and content. Takes one of the following (System is the default):

- a. System: the strategy, mode, content, and coverage are delivered as set by the parameter values;
- b. Learner: the learner may select alternate strategies, mode, content, and coverage.

SLOT: Tactical Control. Determines control over the initiation and termination of interactions. Also determines whether the learner may alter the values of the Guidance, View, and Sequence

parameters. Takes one of two values, System or Learner. The default is System.

3.3.3 KB: Enterprise Knowledge Base

3.3.3.1 FRAME: Enterprise Frame

FRAME: Enterprise Transaction Frame

SLOT: Name
SLOT: Description
SLOT: Parent
SLOT: Children
SLOT: Kind
SLOT: Focus TRX
SLOT: Sequence
 FACET: Primary Sequence
 FACET: Secondary Sequence
 FACET: Combined Sequence
SLOT: Case
 FACET: Class (Optional)
 FACET: Sort_Attribute
SLOT: PreRequisites
SLOT: Elaborations

SLOT: Name. This slot specifies the name of the enterprise.

SLOT: Description. This slot describes the enterprise.

SLOT: Parent. This slot points to the parent enterprise frame if the enterprise is a sub-enterprise.

SLOT: Children. This slot lists the children, if any, of the enterprise. Each member of the list specifies (1) the kind of frame, which can be an enterprise or a transaction frame, and (2) the pointer or link to the particular frame.

SLOT: Kind. Different types of enterprises can be discriminated on the basis of the level of performance required and the type of knowledge involved with the performance. There are six classes of enterprises: denote, evaluate, execute, design, interpret, and discover. This class structure may also be used to classify the enterprise transactions according to the class of enterprise being facilitated.

- a. Denote: A Denote enterprise transaction requires as a focus one of the following: a primary transaction from the Component class, either an Identify transaction for an entity, an Execute transaction at the Denote level of performance for an activity, or an Interpret transaction at the Denote level of performance for a process frame; or a Classify/Decide primary transaction from the Abstraction class. (Performance level is a parameter to Execute and Interpret transaction shells. For Execute, the values may be either Denote or Perform; for Interpret, values are

either Denote, Explain, or Predict.) Primary transactions from the component, abstraction, and association classes will be included to support the focus transaction. Performance for a denote enterprise is characterized as knowing about something. With a Component class primary transaction as focus, the enterprise transaction enables the student to describe the parts, their functions and locations for an entity; describe the steps for an activity; or describe the events for a process. With a Classify/Decide primary transaction as focus, the enterprise transaction enables the student to identify instances or discriminate kinds.

- b. Evaluate: An Evaluate enterprise transaction requires as focus a Judge primary transaction from the Abstraction class. The Judge transaction instructs an abstraction hierarchy of entities, activities, or processes. Primary transactions from the component, abstraction, and association classes will be included to support the focus transaction. Performance for an Evaluate enterprise is characterized as classifying and ranking the adequacy of an entity, the performance of an activity, or the effectiveness of a process.
- c. Execute: An Execute enterprise transaction requires as a focus an Execute primary transaction (at the Perform level) from the Component class. The content for the focus transaction is the steps of an activity frame. Primary transactions from the component, abstraction, and association classes will be included to support the focus transaction. Performance for an Execute enterprise is characterized as performing some activity.
- d. Design: A Design enterprise transaction requires as a focus a Design primary transaction from the Association class. Primary transactions from the component, abstraction, and association classes will be included to support the focus transaction. Performance for a Design enterprise is characterized as inventing or creating a new artifact. It enables the student to design a new entity or activity not previously instructed.
- e. Interpret: An Interpret enterprise transaction requires as a focus an Interpret primary transaction, at the Explain or Predict level of performance, from the Component class. The content for the focus transaction is the events and causal network of a process frame. Primary transactions from the component, abstraction, and association classes will be included to support the focus transaction. Performance for an Interpret enterprise is characterized as knowing why some process works.

- f. Discover: A Discover enterprise transaction requires as a focus a Discover primary transaction from the Association class. Primary transactions from the component, abstraction, and association classes will be included to support the focus transaction. Performance for a Discover enterprise is characterized as finding a new relationship or process. It enables the student to discover a new entity or process not previously instructed.

SLOT: Focus Transaction. This slot specifies the focus transaction of the enterprise. The focus knowledge of the enterprise is specified in the focus transaction.

SLOT: Sequence. There are two dimensions of sequencing at the enterprise level, yielding seven sequencing alternatives.

FACET: Primary Sequence. The first dimension, Primary Sequence, includes Encyclopedic, Case Study, and Situational.

- a. The encyclopedic sequence systematically calls each primary transaction to instruct elements of the content, eventually integrating these at the enterprise level. This type of sequencing is often found in textbooks and reference manuals.
- b. The case study sequence presents a sequence of carefully selected examples, scenarios, or cases of the focus transaction and the necessary supporting transactions, with each case being complete in and of itself. The sequence of cases is graded on some dimension, such as familiarity, frequency, or criticality.
- c. The situational sequence is characterized as on-the-job learning, where instruction is delivered on an as-needed basis. Only that instruction necessary to the immediate task is presented; integration must occur opportunistically. The situational sequence is facilitated by an online advisor system and student modeling.

FACET: Secondary Sequence. The second dimension, which we call Secondary Sequence, includes Elaboration, Prerequisite, and Flat sequence.

- a. Elaboration sequence starts with a simple, representative element or elements of the focus content, and progressively adds layers of detail as the instruction progresses. This is similar in many respects to Riegeluth's Elaboration Theory.
- b. Prerequisite sequence orders elements of subject matter based on their dependency interrelations. This is based on Gagne's learning hierarchies. The focus content is at the top level of the hierarchy.

- c. Flat sequence involves no systematic ordering at the secondary level.

FACET: Combined Sequence. The primary and secondary sequences may be combined into seven approaches to sequencing: elaborated, prerequisite, and flat case study; elaborated, prerequisite, and flat encyclopedic; and situational.

- a. Elaborated case study requires the presence of a number of cases of the focus transaction, each complete in and of itself. In our earlier example of a Circuit Functioning enterprise transaction, the focus transaction was an Interpret primary transaction to instruct circuit functioning. Suppose that the specific enterprise involved the functioning of AC circuits. The enterprise would require the presence of a set of cases, each of which would be instructed by the focus transaction. The cases would be drawn from an abstraction hierarchy of circuits, and would be ordered on some relevant dimension, such as complexity, familiarity, frequency of occurrence, etc. Examples might include specific instances of capacitance reactive circuits, resonant circuits, and transformers. Each case would be an instance of a class in the abstraction hierarchy. However, instructing the abstraction hierarchy is not the focus. The instructing of the abstraction hierarchy is supporting instruction of the focus.

As each case is selected in turn, it is introduced by the focus transaction to the student, following an elaboration secondary sequence. Other information would then be brought into the instruction, from the focus and from supporting transactions, until the circuit had been fully instructed. The next case would then be presented, refreshing and reviewing content that had already been introduced in earlier cases, and introducing additional content.

- b. Prerequisite case study selects cases equivalently, but the secondary sequence follows a prerequisite hierarchy. The case would be overviewed by the focus transaction, but then instruction would build bottom-up following the prerequisites. Each supporting transaction might be called one or more times at different nodes in the hierarchy. Then the next case would be handled in a similar way, refreshing and reviewing content that had already been introduced in earlier cases, and introducing additional content.
- c. Flat case study has no systematic secondary ordering. Once a case had been selected, instruction begins with an overview from the focus, then each supporting transaction will be called in turn to present all required content for that case, finally returning to the focus for a full presentation. Then the next case will be selected.

The encyclopedic sequences are not built on cases. Any abstraction hierarchy is taught as part of the supporting content, rather than being used to generate cases.

- d. Elaborated encyclopedic sequence begins with a representative element or elements of the focus content, introduces supporting content as needed, then builds to the full focus content.
- e. Prerequisite encyclopedic sequence begins with an overview of the focus content, then goes to the lowest levels of prerequisite hierarchy and sequences the primary transactions to deliver instruction for nodes on the hierarchy, building eventually to full focus transaction.
- f. Flat encyclopedic sequence begins with an overview of the focus, then each supporting transaction is called in turn to present all required supporting content, finally returning to the focus for a full presentation.
- g. Situational sequencing delivers instructional elements on demand, either as a result of user request or based on an online determination by an advisor program of the learning requirements of the user.

SLOT: Case

FACET: Class (Optional). If a case sequence is selected, this slot points to the abstraction frame in the DOMAIN.KB which specifies the cases that can be applied during the enterprise.

FACET: Sort Attribute.

SLOT: PreRequisites.

SLOT: Elaborations.

3.3.4 KB: Student Knowledge Base

3.3.4.1 FRAME: Student Profile Frame

FRAME: Student Profile Frame

SLOT: Name
SLOT: Description
SLOT: Control
SLOT: Motivation
SLOT: Familiarity
SLOT: Ability
SLOT: Roles

SLOT: Name. This slot specifies the name of the kind of students profiled by the frame.

SLOT: Description. This slot describes the kind of students profiled by the frame.

SLOT: Control. This slot specifies the focus of control which may be Internal or External.

SLOT: Motivation. This slot specifies the motivation of the student to take the course. The possible values are High, Medium or Low.

SLOT: Familiarity. This slot specifies the student's familiarity with the subject matter. The possible values are High, Medium or Low.

SLOT: Ability. This slot specifies the student's ability/aptitude with regard to the subject matter. The possible values are High, Medium or Low.

SLOT: Roles. This slot specifies the role of the student with regard to the subject matter. The possible values are Consumer, Supervisor, Technician and Problem Solver.

3.3.5 KB: Environment Knowledge Base

3.3.5.1 FRAME: Environment Profile Frame

FRAME: Environment Profile Frame

SLOT: Name
SLOT: Description
SLOT: Delivery Medium
SLOT: Location
SLOT: Schedule
SLOT: Grouping
SLOT: Instructor

SLOT: Name. This slot specifies the name of the environment profiled by the frame.

SLOT: Description. This slot describes the environment profiled by the frame.

SLOT: Delivery Medium. This slot specifies the delivery medium and resources to be used in a particular course. Some possibilities are Computer-Based Instruction and Interactive Video.

SLOT: Location. This slot specifies the kind of location at which the course will be delivered. The possible values are:

- a. Remote Classroom
- b. Local Classroom
- c. Job Site
- d. Home

SLOT: Schedule. This slot specifies whether the course will be delivered with a Fixed or Flexible schedule.

SLOT: Grouping. The slot specifies whether the course will be delivered in Large or Small groups or Individually. The possible values are:

- a. Large Group
- b. Small Group
- c. Individual

SLOT: Instructor. The slot specifies the availability of an instructor during delivery. The possible values are:

- a. Full-Time
- b. Part-Time
- c. Not Available

3.3.6 KB: Task Knowledge Base

3.3.6.1 FRAME: Task Frame

FRAME: Task Frame
SLOT: Name
SLOT: Description
SLOT: Activity

3.4 Knowledge Base and Library References

Following approval of the ESS for XAIDA, this section will provide a cross reference of the architectural elements of the CSCI (e.g., TLCSCs, LLCSCs, Units) to the architectural elements of the AIDA.KB (e.g., KBs, frames, slots, facets). For each KB element, the CSCI element directly referencing the KB element will be listed and the type of reference (set, used, or both) provided. KB references will be depicted in Table 5: Set/Used Table. For convenience of generation and maintenance, this cross reference will be generated by the approved ESS.

4. UNSPECIFIED IN DI-MCCR-80028

5. UNSPECIFIED IN DI-MCCR-80028

6. NOTES

6.1 Background Information

The background of the AIDA project is outlined in (Mei Associates, Inc.: Specifications for an Advanced Instructional Design Advisor (AIDA) for Computer-Based Training, Final Report, Contract No. F33615-88-C-0003, Task Order No. 0006, 31 July 1990, Section 1). Since this document will be bound separately from the SSS and thus may be read separately, relevant extracts are included in this section.

AIDA was first described by Dr. Michael Spector in the final report for his 1988 Summer Faculty Research Program at ALHRD (formerly AFHRL). That research was conducted under the supervision of Dr. Scott Newcomb, Branch Chief for the Training Technology Branch of the Training Systems Division (ALHRD/IDC).

ALHRD/IDC decided to continue the exploratory development of AIDA under Work Unit 1121-10-43, Computer-Based Training (CBT) Software Development and Technical Support. The AIDA project is primarily a response to the Air Training Command (ATC) MPTN 89-14T, Research and Development of Computer-Based Instruction (CBI).

In a follow-on 1989 Research Initiation Program grant, Spector submitted a further report in which he evaluated the potential role for artificial intelligence (AI) in the instructional design process. He concluded that there appears to be a significant role for expert system technology (EST) in instructional design.

Task 0006 was the first of the two ensuing tasks (including the current Task 0013) in which the concrete application of EST to the development of AIDA was explored in detail.

By the conclusion of Task 0013, consensus had been reached that the transaction shell (TRXS) approach to instructional design should be pursued in the research implementation of the AIDA expert system.

In light of these plans, and after consideration of the comparative advantages of the currently available alternate approaches, Mei Associates, Inc. determined to propose a frame-based approach for the AIDA.KB in this DBDD.

6.2 Glossary

Activity: A group of related actions to be performed by a learner in the fulfilment of a specific task. Examples include operating a device and using a formula to perform a calculation.

Association: A non-hierarchical link between frames in a network of frames.

Attribute: A characteristic of an entity, activity, or process. Attributes are represented by frame slots or by an additional frame linked to the slot in the original entity, activity, or process frame. Default values may be supplied by inheritance.

Base Transaction Shell: One of the (currently) twelve basic building blocks of the ID2 transaction shell approach to instructional design. There is one base transaction shell for each of the Primary Transaction Classes: Analogize, Classify/Decide (Inverse of Generalize), Design, Discover, Execute, Generalize (Inverse of Classify), Identify, Interpret, Judge, Propagate, Substitute, and Transfer.

Component: A single item in a particular structural decomposition of entities, activities, and processes: For an entity, one of the parts that make up that entity. For an activity, one of the steps that comprise that activity. For a process, one of the events occurring during, or one of the causes resulting in, the occurrence of that process. A component may be represented by a slot in its entity, activity, or process frame, or by an additional frame connected to the slot of the same name in the original frame.

Course: The instruction adequate to support the performance of one (or several related) enterprise(s).

Domain Knowledge Base: A set of one or more related networks of frames providing the knowledge specific to a particular course or set of related courses.

Entity: In the most general sense, a thing, such as a device, object, person, creature, place, or symbol.

Enterprise: An integrated human performance.

Enterprise Class: (To be filled in on the basis of the completion of Appendix II).

Expert System: A computer program that uses knowledge and automated inferencing techniques to solve problems or perform tasks that, if performed by humans, would require acknowledged expertise in a specific field.

Expert System Shell: An environment built in a high level language that simplifies the construction of expert systems by providing an inference engine and a KBMS.

Frame: A data structure employed to represent knowledge in an expert system. A frame comprises a named set of one or more related slots and is roughly equivalent to a record in conventional data base terminology.

Facet: A variable or a parameter name designating an attribute (value, constraint, link, etc.) of a slot; equivalent to a data item in conventional data base terminology.

Inheritance: The assignment of default attribute values to one or more attributes of a frame as the result of its being linked to a frame at a higher level in a hierarchy within the network of frames comprising a KB.

Knowledge Base: A named network of related frames viewed as a unit.

Link: An inter-frame connection from one frame to another.

Object: A data structure that contains all the information related to a particular entity. It may be considered a frame with some additional features, such as the ability to contain and invoke methods, and the ability to send and receive messages. Objects can be related to other objects by subclass, instance, and other relations, just as frames can.

Primary Transaction Class: One of the (currently) twelve collections of transaction instances by means of which most enterprises may be acquired. Each transaction class corresponds to a single base transaction shell. The current primary transaction classes are: Analogize, Classify/Decide (Inverse of Generalize), Design, Discover, Execute, Generalize (Inverse of Classify), Identify, Interpret, Judge, Propagate, Substitute, and Transfer.

Process: A group of related actions characteristic of an entity but not strictly performable by a human agent, including physical and social events in the real or imagined world. Examples of processes include the functioning of a device, the transmission of a disease, decertification, cell replication, planetary motion, and evolution.

Propagation: One of the two principal means by which knowledge stored in one frame of a network of frames affects knowledge stored in another frame of the same network. In propagation, as contrasted with inheritance, the link between the two frames is not based upon a hierarchical relationship between them.

Second Generation Instructional Design: The theory developed by M. David Merrill and his associates at Utah State University upon which the transaction shell approach to computer-assisted instruction is based.

Slot: A named set of one or more related facets; equivalent to a data field in conventional data base terminology.

Transaction (TRX): A particular, bounded instructional interchange with a student which facilitates the acquisition of a specific competence.

Transaction Class (TRXC): Cf. "Primary Transaction Class."

Transaction Family (TRXF): The interactions necessary to promote the acquisition of all of the knowledge and skill associated with a given enterprise.

Transaction Manager (TRXM): A high level program that can be configured to call and sequence the primary transactions identified as necessary for a curriculum.

Transaction Shell (TRXS): Cf. "Base Transaction Shell."

Transaction Shell Library: The sub-KB of the AIDA.KB which contains the base transaction shells.

Value: The lowest level item of knowledge stored in a KB.

6.3 Acronyms and Abbreviations

AIDA	Advanced Instructional Design Advisor
AIDA.KB	AIDA Knowledge Base
ALHRD/IDC	Armstrong Laboratory, Human Resources Directorate/IDC
AI	Artificial Intelligence
ATC	Air Training Command
CSCI	Computer Software Configuration Item
DBDD	Data Base Design Document (DI-MCCR-80028)
DID	Data Item Description
EFN	Elaborated Frame Network
ES	Expert System
ESS	Expert System Shell
EST	Expert System Technology
GOMS	Goals, Operators, Methods, Selection method
ID	Instructional Design
ID2	Second Generation Instructional Design
ITS	Intelligent Tutoring System
KARS	Knowledge Acquisition/Representation System
KAS	Knowledge Acquisition System
KBM	Knowledge Base Manager
KBMS	Knowledge Base Management System
KBDL	Knowledge Base Definition Language
KBQL	Knowledge Base Query Language

LLCSC	Lower-Level Computer Software Component
NGOMSL	Natural Goals, Operators, Methods, Selection Language
OOD	Object-Oriented
PCA	Physical Configuration Audit
PTC	Primary Transaction Class
PUPS	Penultimate Production System
RAPIDS	Rapid Prototyping Intelligent Tutoring System
SAS	Strategy Analysis System
SME	Subject Matter Expert
SSS	System/Segment Specification
TRX	Transaction
TRXC	Transaction Class
TRXF	Transaction Family
TRXM	Transaction Manager
TRXS	Transaction Shell
TRXS.LIB	Transaction Shell Library
XAIDA	Experimental Advanced Instructional Design Advisor